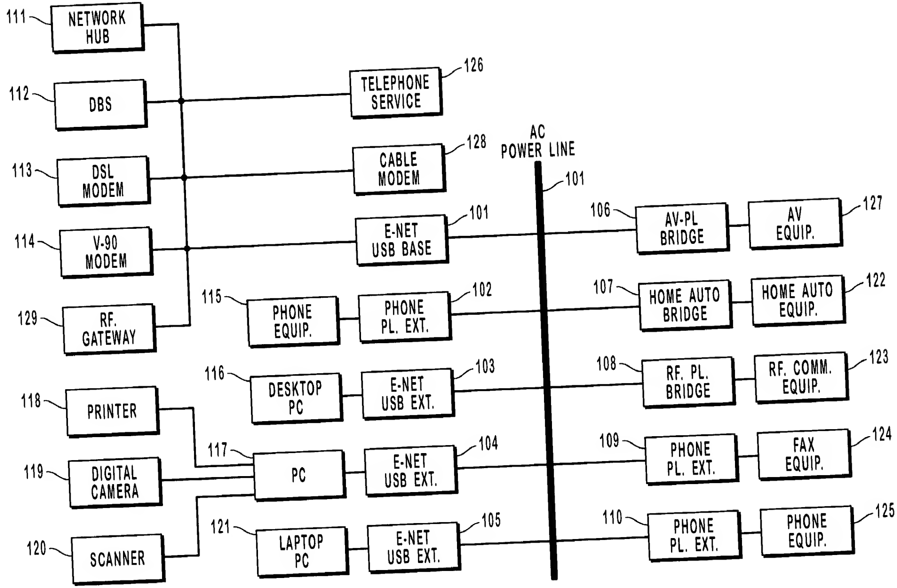

NETWORK HUB
111
DBS
112
DSL MODEM
113
V-90 MODEM
114
RF. GATEWAY
129
PRINTER
118
DIGITAL CAMERA
119
SCANNER
120
TELEPHONE SERVICE
126
CABLE MODEM
128
E-NET USB BASE
101
PHONE EQUIP.
115
PHONE PL. EXT.
102
DESKTOP PC
116
E-NET USB EXT.
103
PC
117
E-NET USB EXT.
104
LAPTOP PC
121
E-NET USB EXT.
105
AC POWER LINE
101
AV-PL BRIDGE
106
AV EQUIP.
127
HOME AUTO BRIDGE
107
HOME AUTO EQUIP.
122
RF. PL. BRIDGE
108
RF. COMM. EQUIP.
123
PHONE PL. EXT.
109
FAX EQUIP.
124
PHONE PL. EXT.
110
PHONE EQUIP.
125

FIG. 1

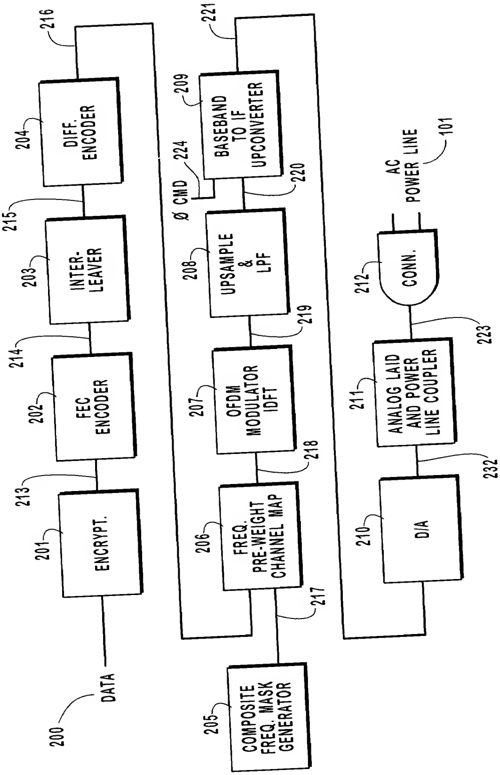

200
DATA
201
ENCRYPT.
213
202
FEC ENCODER
214
203
INTER-LEAVER
215
204
DIFF. ENCODER
216
205
COMPOSITE FREQ. MASK GENERATOR
217
206
FREQ. PRE-WEIGHT CHANNEL MAP
218
207
OFDM MODULATOR IDFT
219
208
UPSAMPLE & LPF
220
Ø CMD
224
209
BASEBAND TO IF UPCONVERTER
221
210
D/A
232
211
ANALOG LAID AND POWER LINE COUPLER
223
212
CONN.
AC POWER LINE
101

FIG. 2

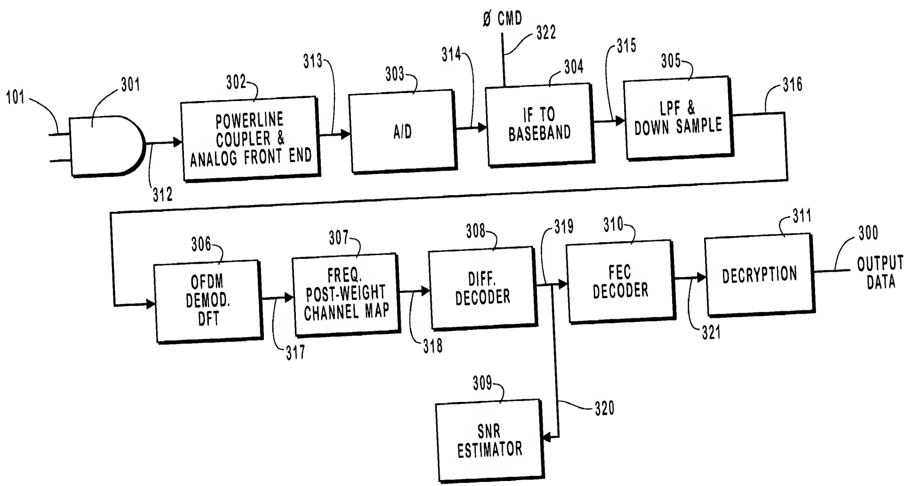

101
301
302
POWERLINE COUPLER & ANALOG FRONT END
312
313
303
A/D
314
Ø CMD
322
304
IF TO BASEBAND
315
305
LPF & DOWN SAMPLE
316
306
OFDM DEMOD. DFT
317
307
FREQ. POST-WEIGHT CHANNEL MAP
318
308
DIFF. DECODER
319
310
FEC DECODER
321
311
DECRYPTION
300
OUTPUT DATA
309
SNR ESTIMATOR
320

FIG. 3

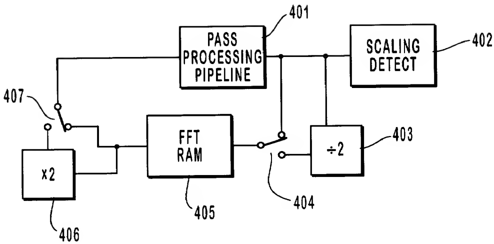

PASS PROCESSING PIPELINE
401
SCALING DETECT
402
407
FFT RAM
÷2
403
x2
405
404
406

FIG. 4

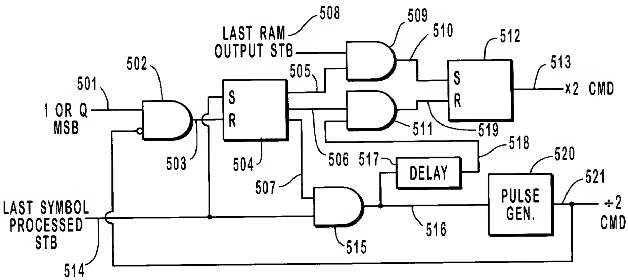

508
LAST RAM OUTPUT STB
509
510
512
502
505
513
501
S
S
x2 CMD
I OR Q MSB
R
R
511
519
518
503
504
506
517
DELAY
520
507
521
LAST SYMBOL PROCESSED STB
PULSE GEN.
÷2 CMD
516
515
514

FIG. 5

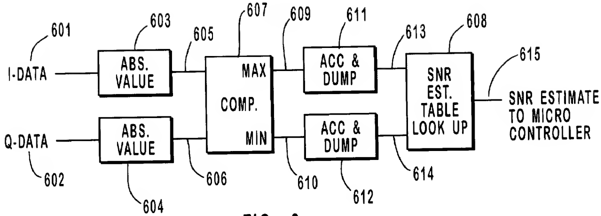


FIG. 6

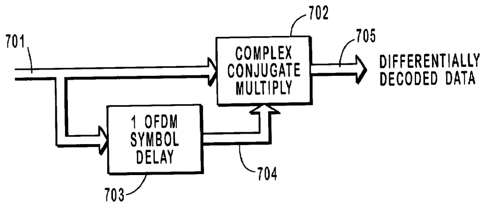


FIG. 7

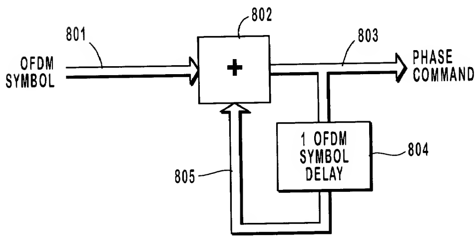


FIG. 8

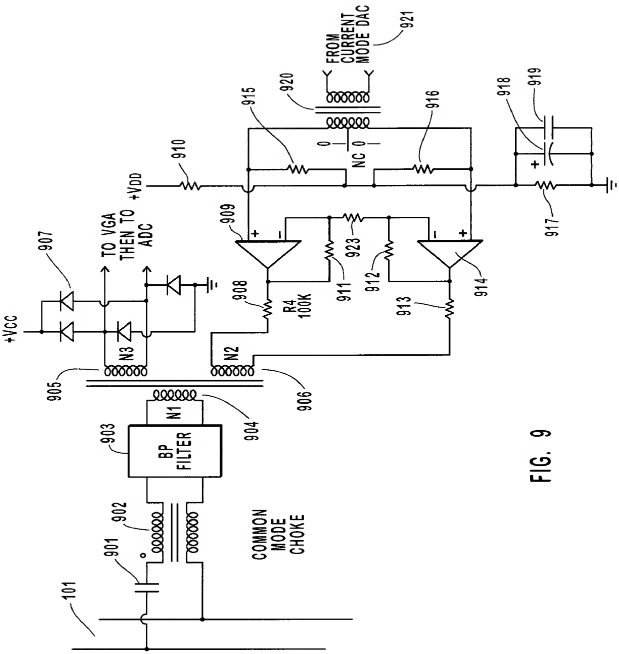

101
901
902
COMMON MODE CHOKE
903
BP FILTER
N1
904
905
906
N3
N2
+VCC
907
TO VGA THEN TO ADC
908
R4 100K
909
911
912
913
914
923
+VDD
910
915
916
920
FROM CURRENT MODE DAC
921
NC
917
918
919

FIG. 9

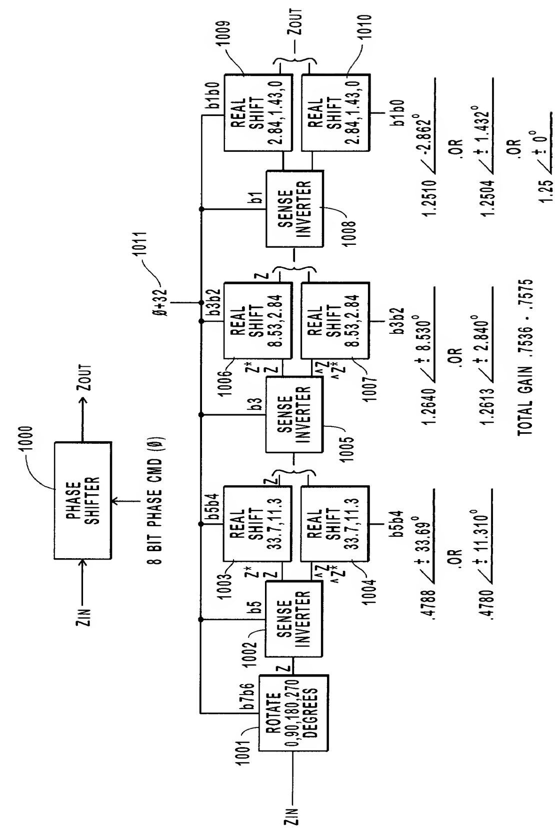

PHASE SHIFTER
1000
ZIN
ZOUT
8 BIT PHASE CMD (Ø)
ROTATE 0,90,180,270 DEGREES
1001
b7b6
SENSE INVERTER
1002
b5
REAL SHIFT 33.7,11.3
1003
1004
b5b4
Z
Z*
^Z
^Z*
SENSE INVERTER
1005
b3
REAL SHIFT 8.53,2.84
1006
1007
b3b2
Ø+32
1011
SENSE INVERTER
1008
b1
REAL SHIFT 2.84,1.43,0
1009
1010
b1b0
ZOUT
.4788 ∠ ± 33.69°
OR
.4780 ∠ ± 11.310°
1.2640 ∠ ± 8.530°
OR
1.2613 ∠ ± 2.840°
1.2510 ∠ -2.862°
OR
1.2504 ∠ ± 1.432°
OR
1.25 ∠ ± 0°
TOTAL GAIN .7536 - .7575

FIG. 10

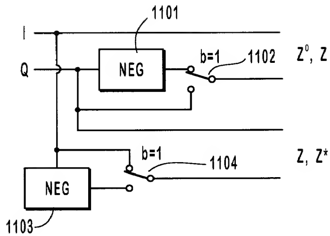

1101
I
Q
NEG
b=1
1102
Z°, Z
b=1
1104
Z, Z*
NEG
1103

FIG. 11

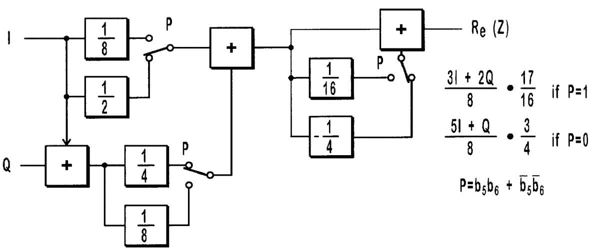
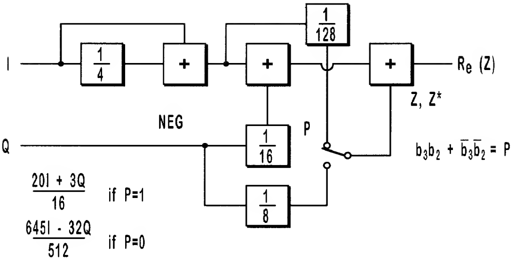
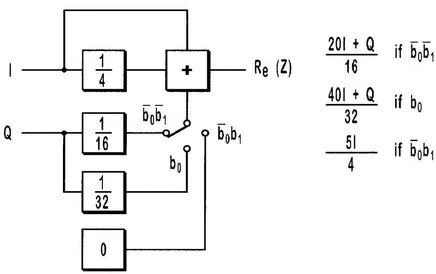

I
1/8
1/2
P
+
Q
+
1/4
1/8
P
1/16
-1/4
P
+
Re (Z)
$\frac{3I + 2Q}{8} \bullet \frac{17}{16}$ if P=1
$\frac{5I + Q}{8} \bullet \frac{3}{4}$ if P=0
$P = b_5 b_6 + \overline{b_5}\,\overline{b_6}$

FIG. 12



1/128
I
1/4
+
+
+
Re (Z)
Z, Z*
NEG
P
Q
1/16
b3b2 + b̄3b̄2 = P
20I + 3Q / 16
if P=1
1/8
645I - 32Q / 512
if P=0

FIG. 13



I
1/4
+
Re (Z)
20I + Q / 16
if b̄0b̄1
40I + Q / 32
if b0
5I / 4
if b̄0b1
b̄0b̄1
Q
1/16
b̄0b1
b0
1/32
0

FIG. 14

```
function z= pshift1(z,th);
iz=real(z); qz=imag(z);
tha=2*(bitand(th+32,2.^(0:7))>0)-1; %make array of phase bits + 32

%0,90,180,270
if tha(7)= =1;z=j*z;end;
if tha(8)= =1;z= -z; end;

% + - 33.69,11.31
if tha(6)*tha(5)= = -1;
        z=3/16*(5+tha(6)*j)*z;
else;
        z=17/64*(3+tha(6)*2*j)*z;
end;

% +- 8.53,2.84
if tha(4)*tha(3)= =-1;
        z=(20*(1+1/128) + tha(4)*j)*z/32;
else;
        z=(20+tha(4)*3*j)*z/32;
end;

% -2.86 –1.43 0 1.43
if tha(1)= =1;
        z=(40+tha(2)*j)*z/32;
elseif tha(2)= = -1;
        z=(20-j)*z/16;
else;
        z=5*z/4;
end;
```

FIG. 15

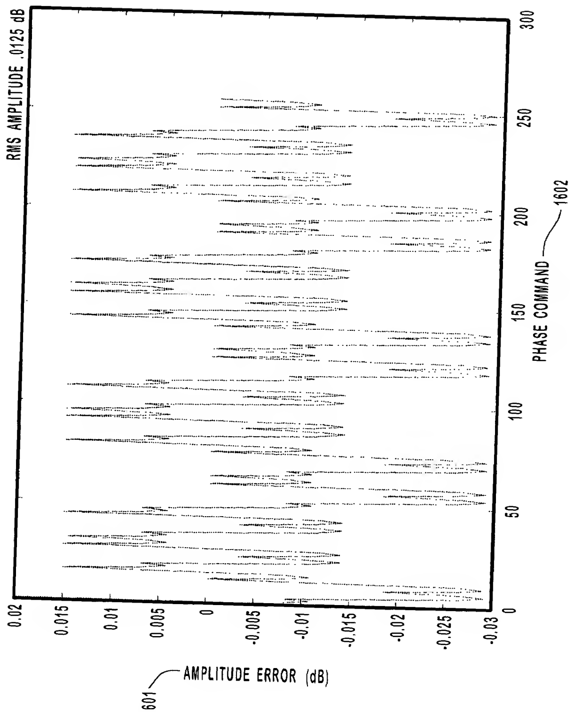

RMS AMPLITUDE .0125 dB
AMPLITUDE ERROR (dB)
1601
PHASE COMMAND
1602
0.02
0.015
0.01
0.005
0
-0.005
-0.01
-0.015
-0.02
-0.025
-0.03
0
50
100
150
200
250
300

FIG. 16

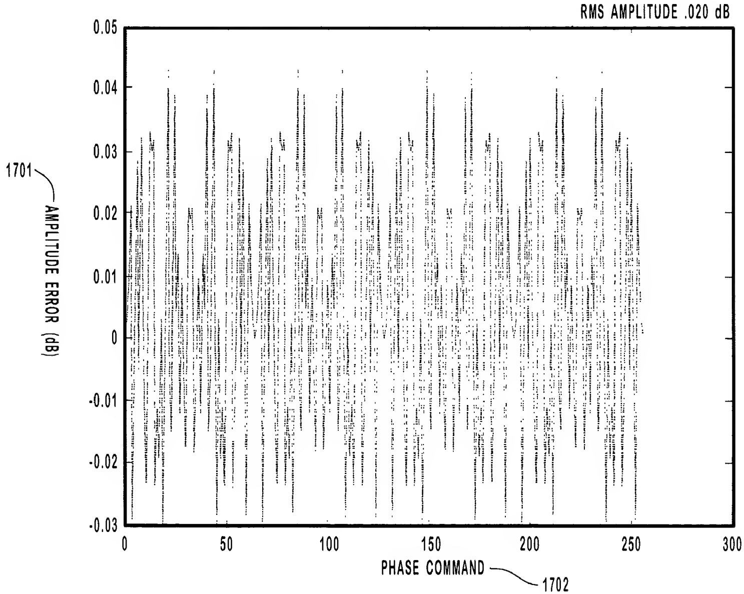

RMS AMPLITUDE .020 dB
0.05
0.04
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
1701
AMPLITUDE ERROR (dB)
0
50
100
150
200
250
300
PHASE COMMAND
1702

FIG. 17

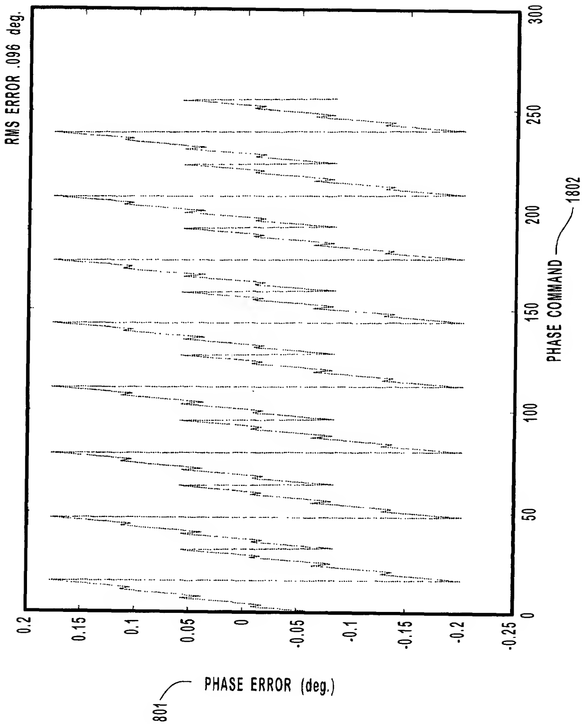

RMS ERROR .096 deg.
PHASE ERROR (deg.)
1801
PHASE COMMAND
1802
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
0
50
100
150
200
250
300

FIG. 18

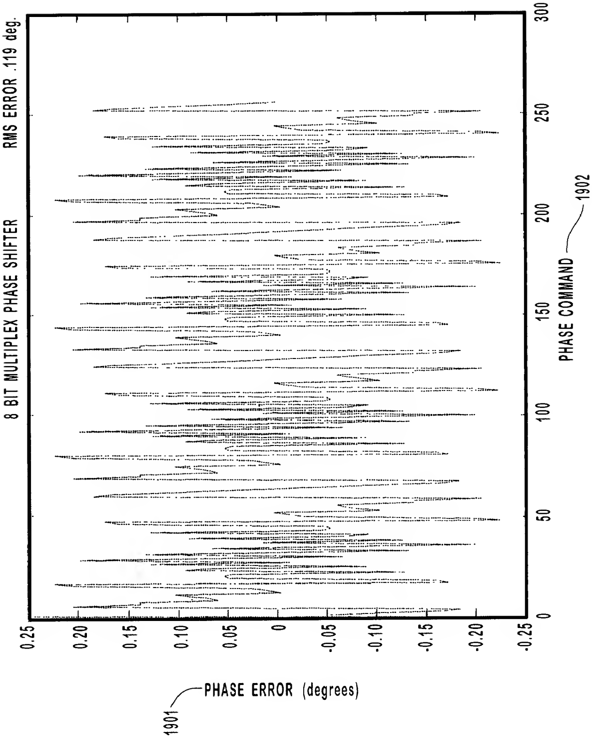
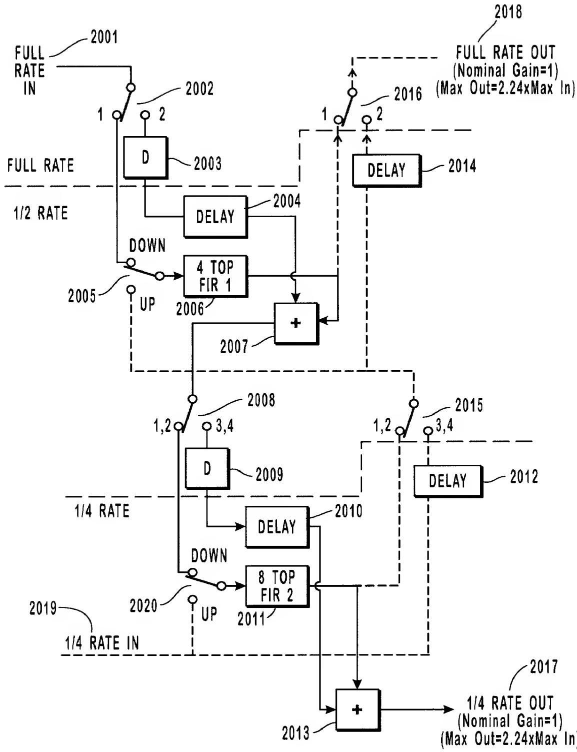

8 BIT MULTIPLEX PHASE SHIFTER
RMS ERROR .119 deg.
PHASE ERROR (degrees)
1901
0.25
0.20
0.15
0.10
0.05
0
-0.05
-0.10
-0.15
-0.20
-0.25
PHASE COMMAND
1902
0
50
100
150
200
250
300

FIG. 19



2018
FULL RATE OUT
(Nominal Gain=1)
(Max Out=2.24xMax In)
2001
FULL
RATE
IN
2002
1
2
2016
1
2
FULL RATE
D
2003
DELAY
2014
1/2 RATE
DELAY
2004
DOWN
4 TOP
FIR 1
2005
UP
2006
+
2007
2008
1,2
3,4
2015
1,2
3,4
D
2009
DELAY
2012
1/4 RATE
DELAY
2010
DOWN
8 TOP
FIR 2
2019
2020
UP
2011
1/4 RATE IN
2017
+
1/4 RATE OUT
(Nominal Gain=1)
(Max Out=2.24xMax In)
2013

FIG. 20

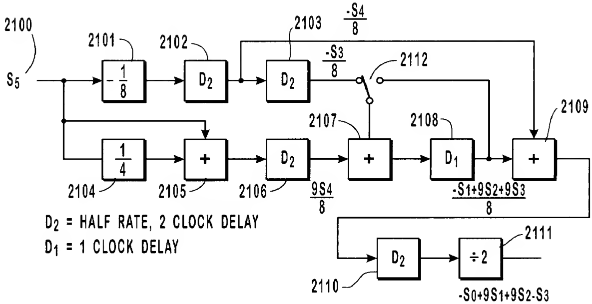

2100
S5
2101
- 1/8
2102
D2
2103
D2
-S4/8
-S3/8
2112
2109
2107
2108
1/4
+
D2
+
D1
+
2104
2105
2106
9S4/8
-S1+9S2+9S3/8
D2 = HALF RATE, 2 CLOCK DELAY
D1 = 1 CLOCK DELAY
2111
D2
÷2
2110
-S0+9S1+9S2-S3

FIG. 21

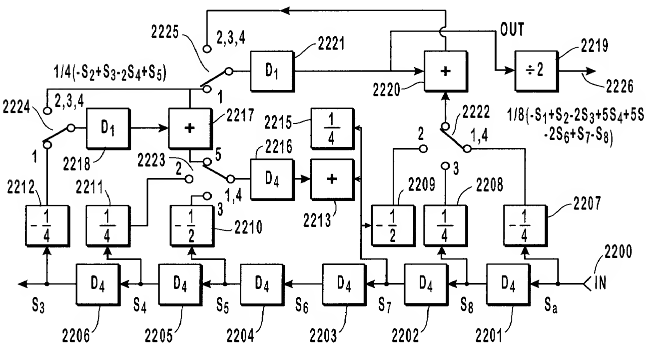

2225
2,3,4
2221
OUT
D1
+
÷2
2219
1/4(-S2+S3-2S4+S5)
2220
2226
2224
2,3,4
1
D1
+
2217
2215
1/4
2222
1/8(-S1+S2-2S3+5S4+5S5
-2S6+S7-S8)
1
2218
2223
5
2216
2
1,4
2
D4
+
3
1,4
2212
2211
2209
2208
2207
3
2213
- 1/4
1/4
- 1/2
2210
- 1/2
1/4
- 1/4
2200
D4
D4
D4
D4
D4
D4
IN
S3
S4
S5
S6
S7
S8
Sa
2206
2205
2204
2203
2202
2201

FIG. 22

Anatomy of the FFT

This is a 256 point base 2 example of an FFT (a fast dft)

$k := 0..255$ FFT index

$x_k := \mathrm{rnd}(1) + j \cdot \mathrm{rnd}(1)$ FFT dummy arguement

The discfrete forrier transform takes 256 ^2 operations

$$DFT_k := \sum_{i=0}^{255} x_i \cdot e^{j \cdot 2 \cdot \pi \cdot \frac{i \cdot k}{256}}$$

If we combine terms from the first and 2nd half of the summation we have

$$\sum_{i=0}^{127} \left(x_i + x_{i+128} \cdot e^{j \cdot 2 \cdot \pi \cdot \frac{k}{2}} \right) \cdot e^{j \cdot 2 \cdot \pi \cdot \frac{i \cdot k}{256}}$$

Note the term in paremphases runs over half of N and is only unique for k mod 2 so:

$i1 := 0..127 \qquad k1 := 0..1$

$$X1_{i1,k1} := x_{i1} + x_{i1+128} \cdot e^{(j \cdot 2 \cdot \pi) \cdot \frac{k1}{2}}$$

Then

$$DFT1_k := \sum_{i=0}^{127} X1_{i,\mathrm{mod}(k,2)} \cdot e^{j \cdot 2 \cdot \pi \cdot \frac{i \cdot k}{256}} \qquad \sum_k \left(\left| DFT_k - DFT1_k \right| \right)^2 = 0$$

This operation only takes half the DFT steps by taking 128 steps to precompute X1

We can further reduce the computational load by doing the same thing again

$i1 := 0..63 \qquad k1 := 0..3$

$$X2_{i1,k1} := X1_{i1,\mathrm{mod}(k1,2)} + X1_{i1+64,\mathrm{mod}(k1,2)} \cdot e^{(j \cdot 2 \cdot \pi) \cdot \frac{k1}{4}}$$

Then

$$DFT2_k := \sum_{i=0}^{63} X2_{i,\mathrm{mod}(k,4)} \cdot e^{j \cdot 2 \cdot \pi \cdot \frac{i \cdot k}{256}} \qquad \sum_k \left(\left| DFT_k - DFT2_k \right| \right)^2 = 0$$

FIG. 23a

We repeat this operation a total of 8 times until i ranges only over zero
We show one more trick. Instead of:

$$i1 := 0..31 \qquad k1 := 0..7$$

$$X3_{i1,k1} := X2_{i1,\mathrm{mod}(k1,4)} + X2_{i1+32,\mathrm{mod}(k1,4)} \cdot e^{(j \cdot 2 \cdot \pi) \cdot \frac{k1}{8}}$$

Use

$$k1 := 0..3 \qquad t_{i1,k1} := X2_{i1+32,k1} \cdot e^{(j \cdot 2 \cdot \pi) \cdot \frac{k1}{8}}$$

$$X3_{i1,k1} := X2_{i1,k1} + t_{i1,k1} \qquad X3_{i1,k1+4} := X2_{i1,k1} - t_{i1,k1}$$

It only requires half the phase shifts. This operation is called a **butterfly**
Continuing

$$i1 := 0..15 \qquad k1 := 0..7 \qquad t_{i1,k1} := X3_{i1+16,k1} \cdot e^{(j \cdot 2 \cdot \pi) \cdot \frac{k1}{16}}$$

$$X4_{i1,k1} := X3_{i1,k1} + t_{i1,k1} \qquad X4_{i1,k1+8} := X3_{i1,k1} - t_{i1,k1}$$

$$i1 := 0..7 \qquad k1 := 0..15 \qquad t_{i1,k1} := X4_{i1+8,k1} \cdot e^{(j \cdot 2 \cdot \pi) \cdot \frac{k1}{32}}$$

$$X5_{i1,k1} := X4_{i1,k1} + t_{i1,k1} \qquad X5_{i1,k1+16} := X4_{i1,k1} - t_{i1,k1}$$

$$i1 := 0..3 \qquad k1 := 0..31 \qquad t_{i1,k1} := X5_{i1+4,k1} \cdot e^{(j \cdot 2 \cdot \pi) \cdot \frac{k1}{64}}$$

$$X6_{i1,k1} := X5_{i1,k1} + t_{i1,k1} \qquad X6_{i1,k1+32} := X5_{i1,k1} - t_{i1,k1}$$

$$i1 := 0..1 \qquad k1 := 0..63 \qquad t_{i1,k1} := X6_{i1+2,k1} \cdot e^{(j \cdot 2 \cdot \pi) \cdot \frac{k1}{128}}$$

$$X7_{i1,k1} := X6_{i1,k1} + t_{i1,k1} \qquad X7_{i1,k1+64} := X6_{i1,k1} - t_{i1,k1}$$

and finally

$$k1 := 0..127 \qquad t_{0,k1} := X7_{1,k1} \cdot e^{(j \cdot 2 \cdot \pi) \cdot \frac{k1}{256}}$$

$$X8_{k1} := X7_{0,k1} + t_{0,k1} \qquad X8_{k1+128} := X7_{0,k1} - t_{0,k1}$$

$$\sum_{k} \left(\left| DFT_k - X8_k \right| \right)^2 = 0$$

FIG. 23b

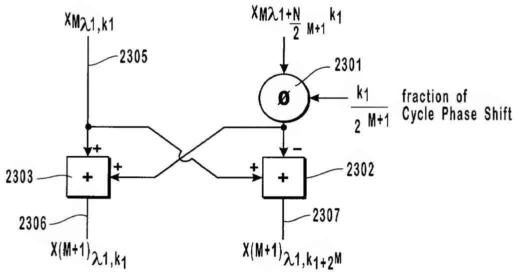


FIG. 23c

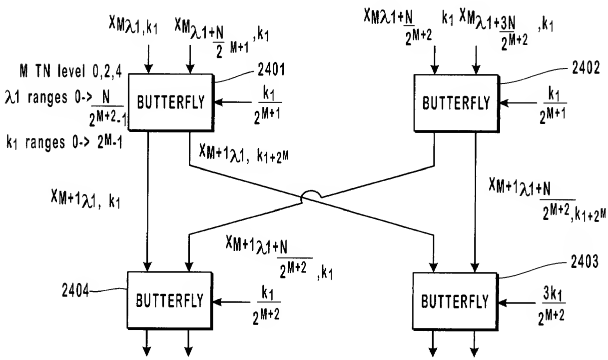


FIG. 24

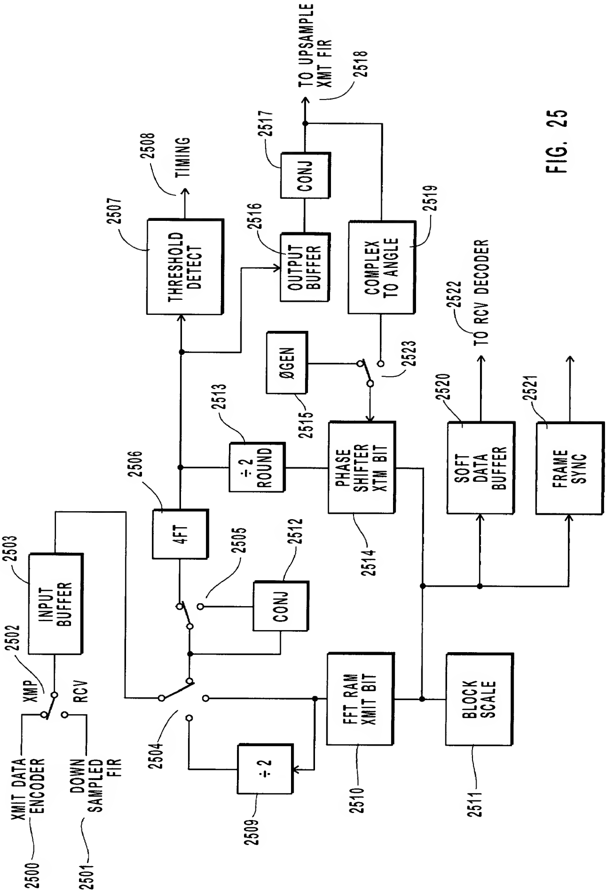
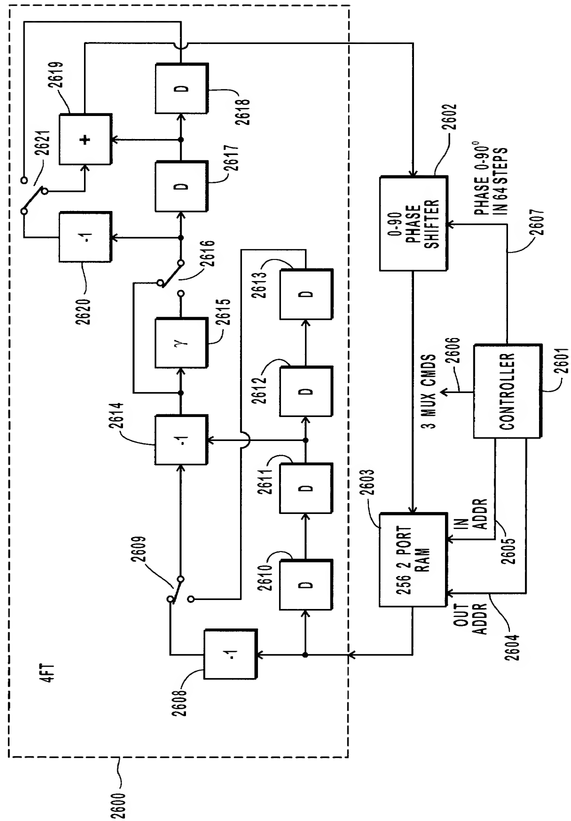

XMIT DATA ENCODER
2500
DOWN SAMPLED FIR
2501
XMP
RCV
2502
INPUT BUFFER
2503
2504
÷ 2
2509
2505
CONJ
2512
4FT
2506
÷ 2 ROUND
2513
THRESHOLD DETECT
2507
TIMING
2508
FFT RAM XMIT BIT
2510
BLOCK SCALE
2511
PHASE SHIFTER XTM BIT
2514
ØGEN
2515
OUTPUT BUFFER
2516
CONJ
2517
TO UPSAMPLE XMT FIR
2518
COMPLEX TO ANGLE
2519
2523
SOFT DATA BUFFER
2520
TO RCV DECODER
2522
FRAME SYNC
2521

FIG. 25



4FT
2600
2608
-1
2609
2610
D
2611
D
2612
D
2613
D
2614
-1
2615
γ
2616
2617
D
2618
D
2619
+
2620
-1
2621
0-90 PHASE SHIFTER
2602
PHASE 0-90° IN 64 STEPS
2607
CONTROLLER
2601
3 MUX CMDS
2606
256 2 PORT RAM
2603
IN ADDR
2605
OUT ADDR
2604

FIG. 26

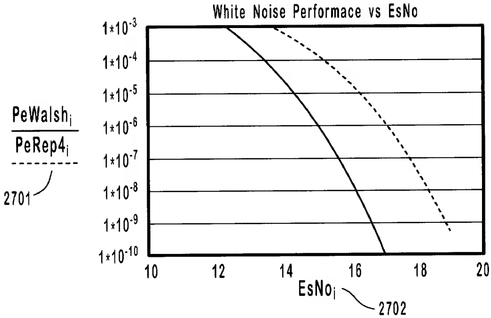


FIG. 27a

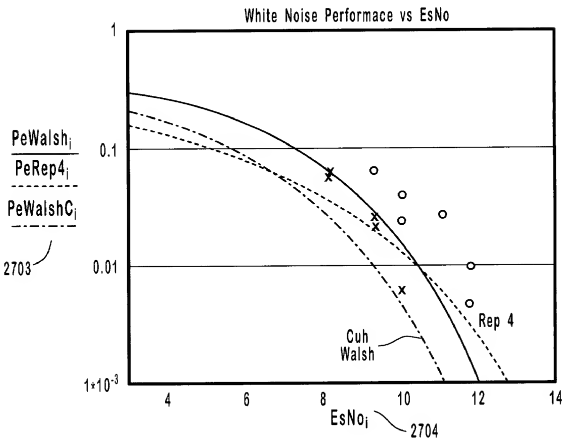


FIG. 27b

```
% finds likelihood of 16ary symbol error for nbe erasures
tr={1 –1; 1 1]; tr=[tr –tr; tr tr]; tr=[tr –tr; tr tr]; tr=[tr –tr; tr tr];
v=zeros(1,16);
nsym=zeros(1,16);
while sum(v)<16;
        [a,n]=max (tr*v');
        if n<16;
                nbe=sum(v= =0);
                nsym(nbe)=nsym(nbe) +1;
        end;
        k=1;
        v(k)= -(v(k)-1); % -1 or 0 – erasures
        while v(k)= =0;
                k=k+1;
                v(k)= -(v(k)-1); % -1 or 0 – erasures
        end;
end;
pe=.005:.005:.5;
for k=1:100;
        we(k)=sum(pe(k).^(1:16).*(1-pe(k)).^(15:-1:0).*nsym);
end;
rhe=1-((1-pe).^4+4*pe.*(1-pe).^3).^4; % repeat sym hard error
rea=1-(1-pe.^4).64; % repeat sym erasure error
plot (pe,we,pe,rea)
```

FIG. 28

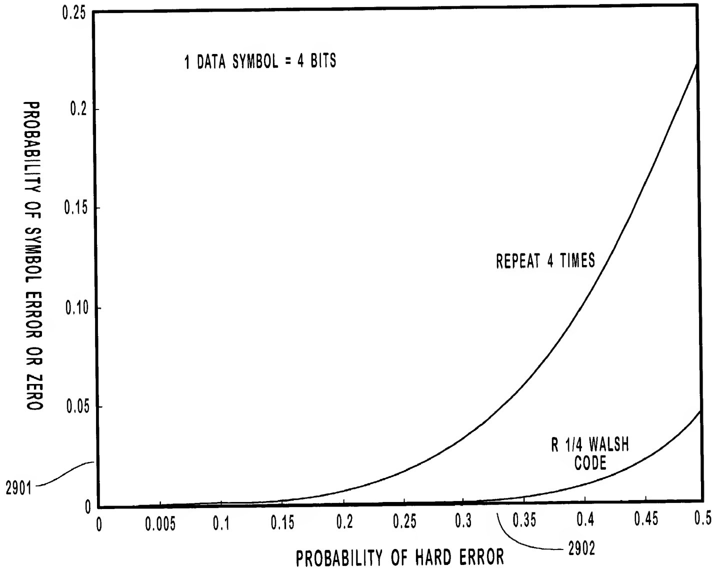

1 DATA SYMBOL = 4 BITS
REPEAT 4 TIMES
R 1/4 WALSH CODE
PROBABILITY OF SYMBOL ERROR OR ZERO
0.25
0.2
0.15
0.10
0.05
0
0
0.005
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
PROBABILITY OF HARD ERROR
2901
2902

FIG. 29

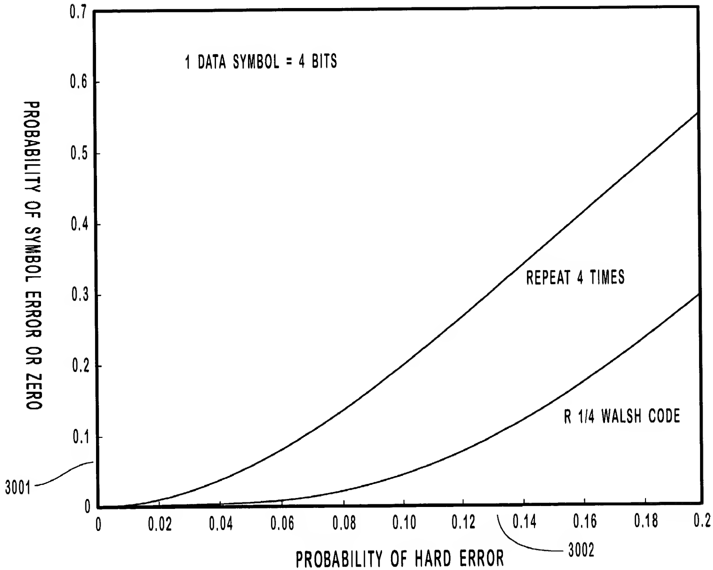

1 DATA SYMBOL = 4 BITS
REPEAT 4 TIMES
R 1/4 WALSH CODE
PROBABILITY OF SYMBOL ERROR OR ZERO
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
PROBABILITY OF HARD ERROR
0
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0.18
0.2
3001
3002

FIG. 30

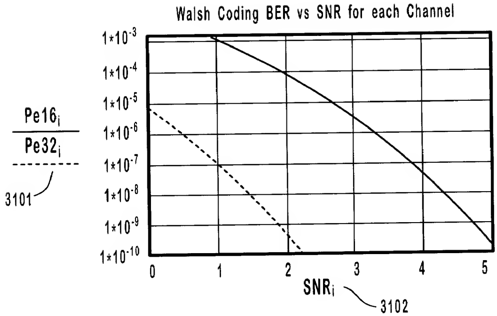

Walsh Coding BER vs SNR for each Channel
1*10^-3
1*10^-4
1*10^-5
1*10^-6
1*10^-7
1*10^-8
1*10^-9
1*10^-10
0
1
2
3
4
5
SNRi
3102
Pe16i
Pe32i
3101

FIG. 31

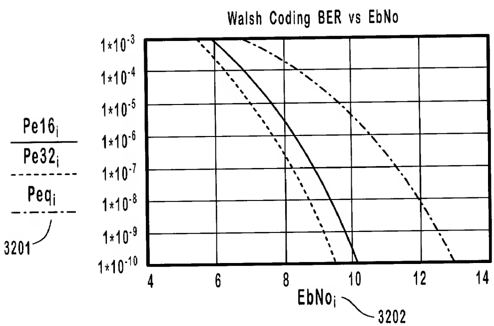

Walsh Coding BER vs EbNo
1*10^-3
1*10^-4
1*10^-5
1*10^-6
1*10^-7
1*10^-8
1*10^-9
1*10^-10
4
6
8
10
12
14
EbNoi
3202
Pe16i
Pe32i
Peqi
3201

FIG. 32

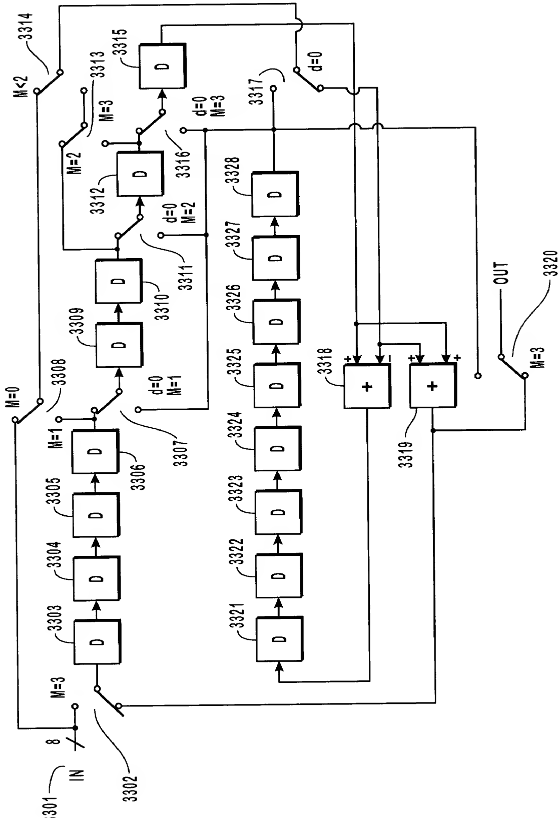

IN
3301
8
M=3
3302
3303
3304
3305
3306
M=1
3307
d=0
M=1
M=0
3308
3309
3310
3311
d=0
M=2
3312
M=2
M=3
3313
M<2
3314
3315
3316
d=0
M=3
3317
d=0
3318
3319
3320
OUT
M=3
3321
3322
3323
3324
3325
3326
3327
3328
D
+

FIG. 33

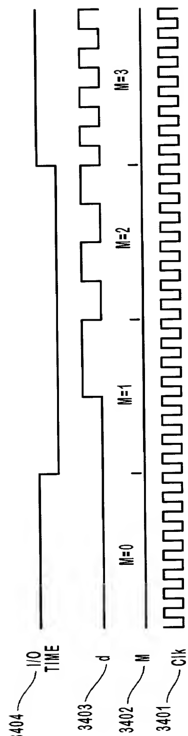


FIG. 34

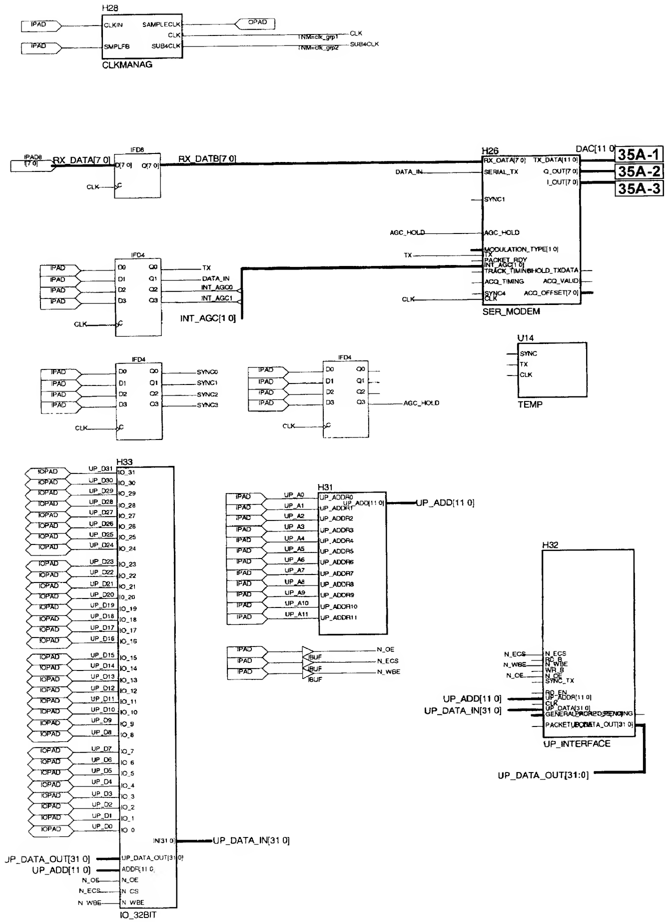

H28
CLKMANAG
CLKIN
SAMPLECLK
CLK
SMPLFB
SUB4CLK
IPAD
OPAD
CLK
SUB4CLK
IFD8
RX_DATA[7 0]
RX_DATB[7 0]
H26
SER_MODEM
DAC[11 0]
35A-1
35A-2
35A-3
DATA_IN
AGC_HOLD
TX
CLK
IFD4
INT_AGC[1 0]
U14
SYNC
TX
CLK
TEMP
SYNC0
SYNC1
SYNC2
SYNC3
AGC_HOLD
H33
IO_32BIT
IOPAD
UP_DATA_IN[31 0]
UP_DATA_OUT[31 0]
UP_ADD[11 0]
N_OE
N_ECS
N_WBE
H31
UP_ADD[11 0]
IBUF
N_OE
N_ECS
N_WBE
H32
UP_INTERFACE
UP_ADD[11 0]
UP_DATA_IN[31 0]
UP_DATA_OUT[31:0]

FIG. 35a-1

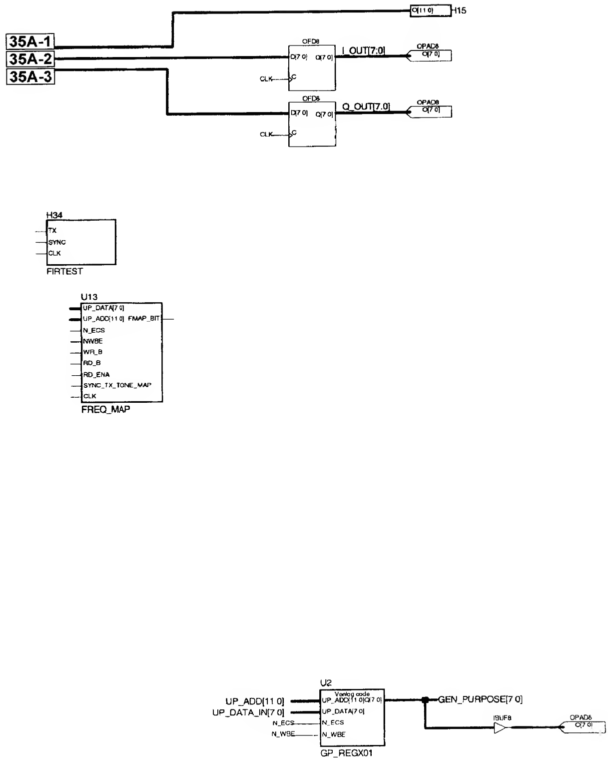

35A-1
35A-2
35A-3
O[11 0]
H15
OFD8
D[7 0]
Q[7 0]
C
CLK
I_OUT[7:0]
OPAD8
O[7 0]
Q_OUT[7:0]
H34
TX
SYNC
CLK
FIRTEST
U13
UP_DATA[7 0]
UP_ADD[11 0]
FMAP_BIT
N_ECS
NWBE
WR_B
RD_B
RD_ENA
SYNC_TX_TONE_MAP
CLK
FREQ_MAP
U2
Verilog code
UP_ADD[11 0]
UP_DATA_IN[7 0]
N_ECS
N_WBE
UP_DATA[7 0]
GEN_PURPOSE[7 0]
IBUF8
OPAD8
O[7 0]
GP_REGX01

FIG. 35a-2

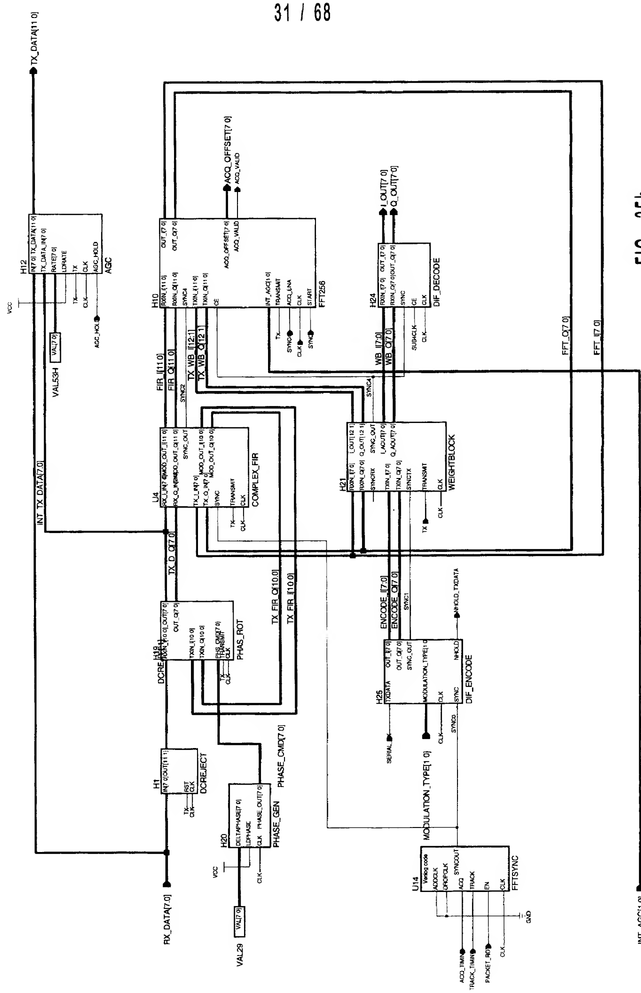

TX_DATA[11:0]
H12
AGC
VCC
VAL53H
AGC_HOLD
INT TX DATA[7:0]
H10
FFT256
ACQ_OFFSET[7:0]
ACQ_VALID
FIR_I[11:0]
FIR_Q[11:0]
SYNC2
TX_WB_I[12:1]
TX_WB_Q[12:1]
FFT_Q[7:0]
FFT_I[7:0]
H24
DIF_DECODE
I_OUT[7:0]
Q_OUT[7:0]
WB_I[7:0]
WB_Q[7:0]
SYNC4
U4
COMPLEX_FIR
H21
WEIGHTBLOCK
TX_D_Q[7:0]
TX_FIR_Q[10:0]
TX_FIR_I[10:0]
H19
DCRE
PHAS_ROT
ENCODE_I[7:0]
ENCODE_Q[7:0]
SYNC1
NHOLD_TXDATA
H25
DIF_ENCODE
SERIAL
MODULATION_TYPE[1:0]
H1
DCREJECT
H20
PHASE_GEN
PHASE_CMD[7:0]
VAL29
U14
FFTSYNC
GND
RX_DATA[7:0]
ACQ_TIMIN
TRACK_TIMIN
PACKET_RO
CLK
INT_AGC[1:0]

FIG. 35b

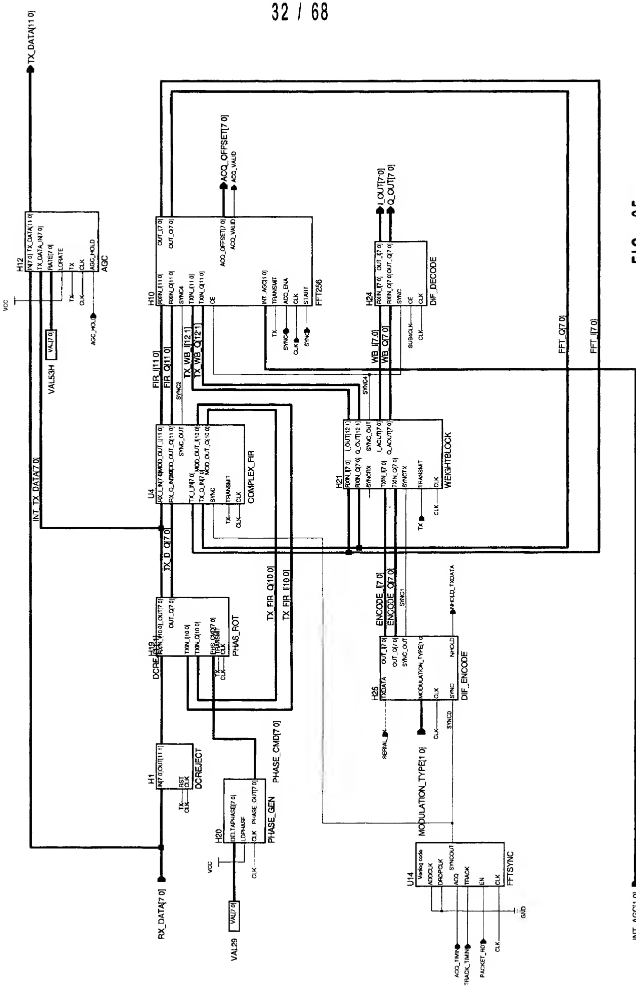

TX_DATA[11 0]
AGC
FFT256
ACQ_OFFSET[7 0]
ACQ_VALID
I_OUT[7 0]
Q_OUT[7 0]
DIF_DECODE
WEIGHTBLOCK
COMPLEX_FIR
PHAS_ROT
DCREJECT
PHASE_GEN
PHASE_CMD[7 0]
DIF_ENCODE
FFTSYNC
MODULATION_TYPE[1 0]
RX_DATA[7 0]
INT_AGC[1 0]
INT_TX_DATA[7 0]
VAL53H
VAL29

FIG. 35c

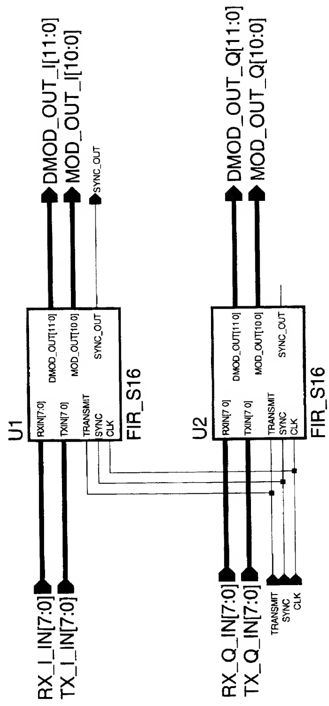

RX_I_IN[7:0]
TX_I_IN[7:0]
U1
RXIN[7:0]
TXIN[7:0]
TRANSMIT
SYNC
CLK
DMOD_OUT[11:0]
MOD_OUT[10:0]
SYNC_OUT
FIR_S16
DMOD_OUT_I[11:0]
MOD_OUT_I[10:0]
SYNC_OUT
RX_Q_IN[7:0]
TX_Q_IN[7:0]
TRANSMIT
SYNC
CLK
U2
RXIN[7:0]
TXIN[7:0]
TRANSMIT
SYNC
CLK
DMOD_OUT[11:0]
MOD_OUT[10:0]
SYNC_OUT
FIR_S16
DMOD_OUT_Q[11:0]
MOD_OUT_Q[10:0]

FIG. 35d

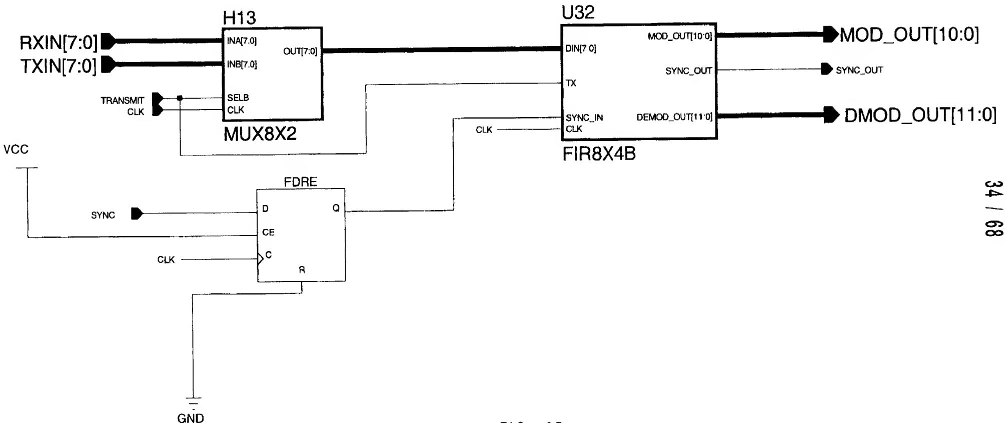

H13
RXIN[7:0]
TXIN[7:0]
INA[7:0]
INB[7:0]
OUT[7:0]
TRANSMIT
CLK
SELB
CLK
MUX8X2
U32
DIN[7:0]
TX
SYNC_IN
CLK
MOD_OUT[10:0]
SYNC_OUT
DEMOD_OUT[11:0]
FIR8X4B
MOD_OUT[10:0]
SYNC_OUT
DMOD_OUT[11:0]
CLK
VCC
FDRE
SYNC
D
CE
C
Q
R
CLK
GND

FIG. 35e

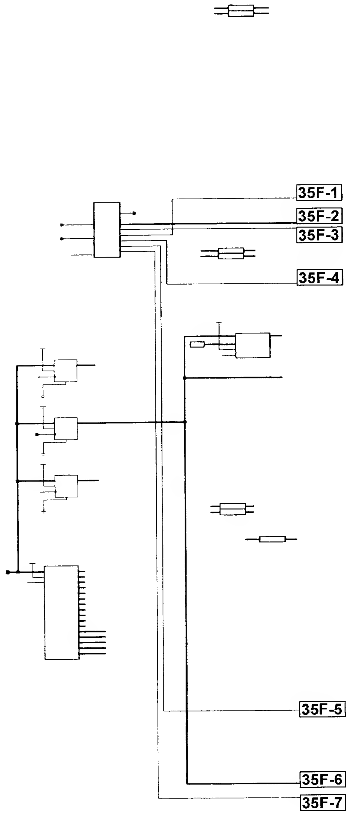


FIG. 35f-1

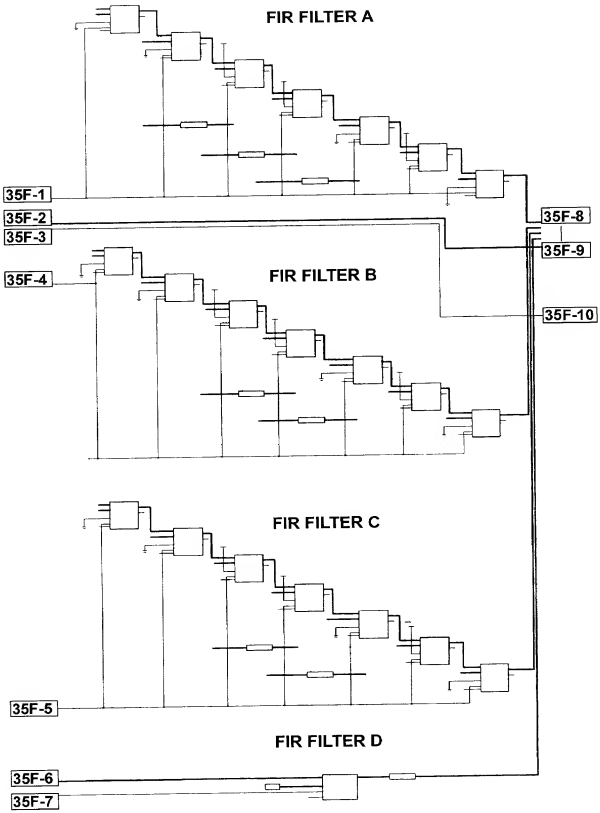

FIR FILTER A
35F-1
35F-2
35F-3
35F-8
35F-9
FIR FILTER B
35F-4
35F-10
FIR FILTER C
35F-5
FIR FILTER D
35F-6
35F-7

FIG. 35f-2

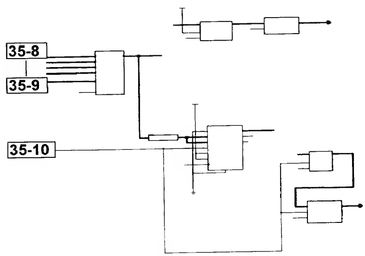


FIG. 35f-3

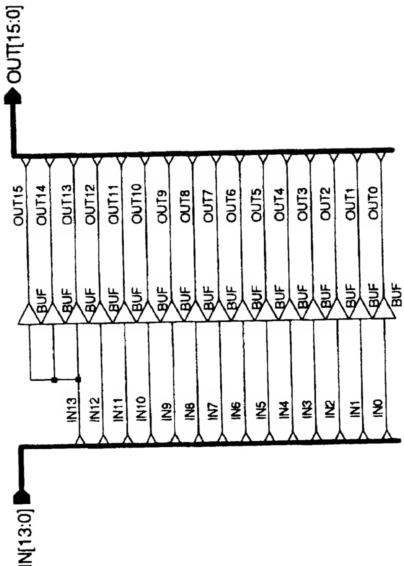

OUT[15:0]
IN[13:0]
OUT15
OUT14
OUT13
OUT12
OUT11
OUT10
OUT9
OUT8
OUT7
OUT6
OUT5
OUT4
OUT3
OUT2
OUT1
OUT0
BUF
IN13
IN12
IN11
IN10
IN9
IN8
IN7
IN6
IN5
IN4
IN3
IN2
IN1
IN0

FIG. 35g

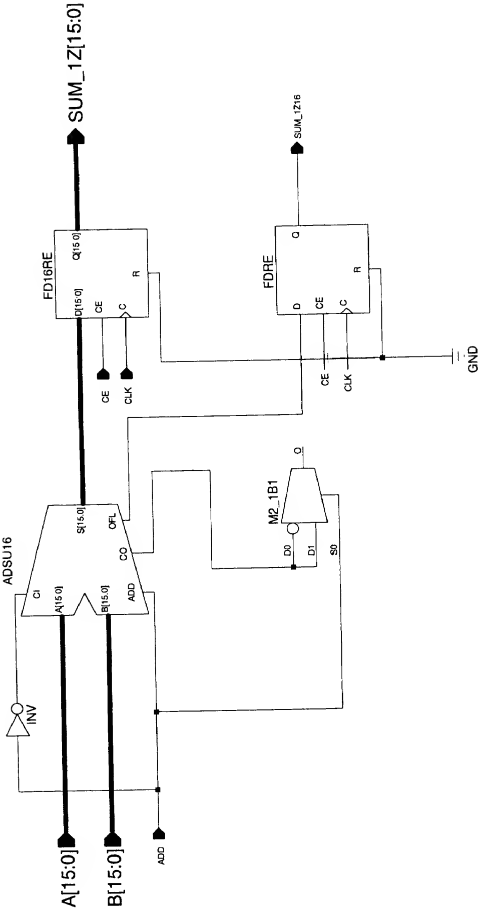

SUM_1Z[15:0]
SUM_1Z16
FD16RE
D[15:0]
Q[15:0]
CE
C
R
FDRE
D
Q
CE
C
R
CE
CLK
CE
CLK
GND
ADSU16
CI
A[15:0]
B[15:0]
ADD
CO
OFL
S[15:0]
M2_1B1
D0
D1
S0
O
INV
A[15:0]
B[15:0]
ADD

FIG. 35h

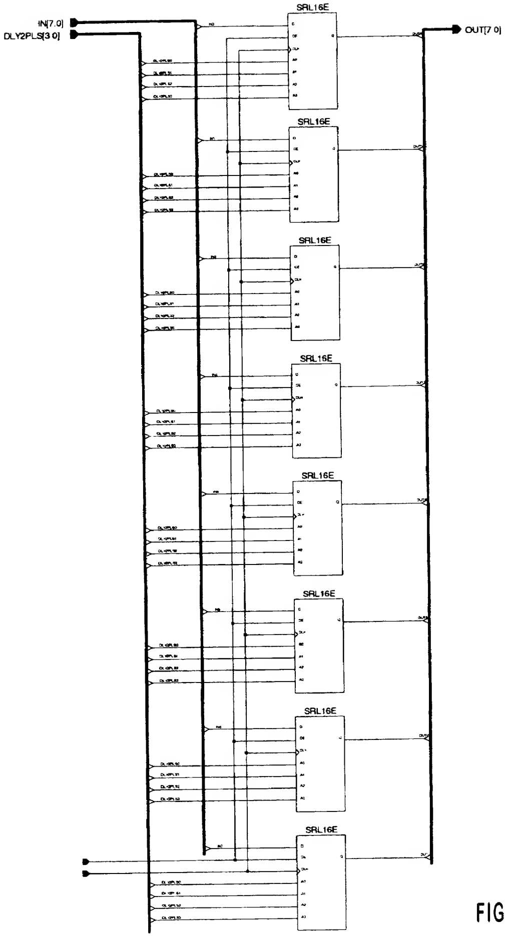

IN[7:0]
DLY2PLS[3:0]
OUT[7:0]
SRL16E
SRL16E
SRL16E
SRL16E
SRL16E
SRL16E
SRL16E
SRL16E

FIG. 35i

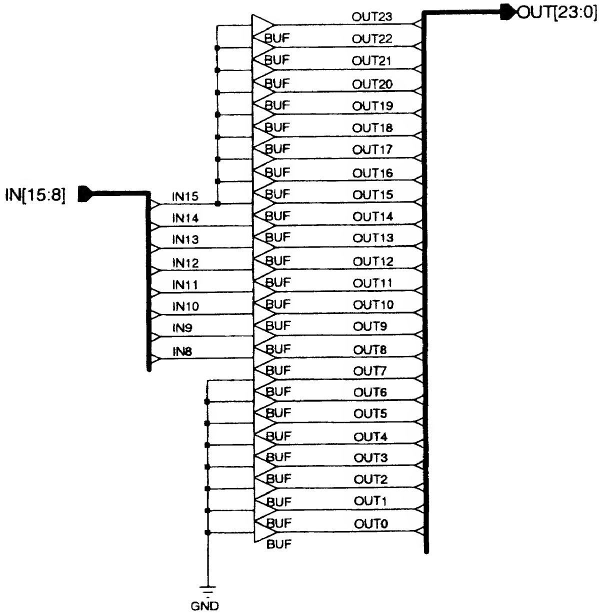

OUT[23:0]
IN[15:8]
IN15
IN14
IN13
IN12
IN11
IN10
IN9
IN8
OUT23
OUT22
OUT21
OUT20
OUT19
OUT18
OUT17
OUT16
OUT15
OUT14
OUT13
OUT12
OUT11
OUT10
OUT9
OUT8
OUT7
OUT6
OUT5
OUT4
OUT3
OUT2
OUT1
OUT0
BUF
GND

FIG. 35j

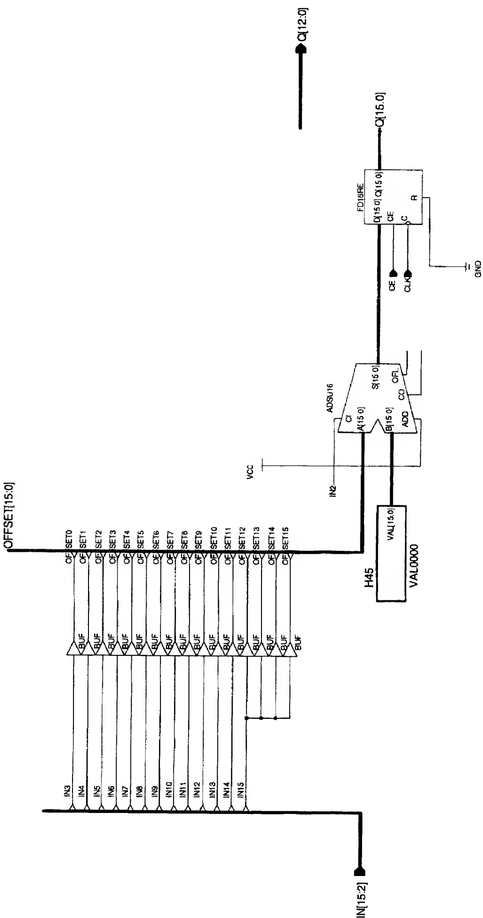
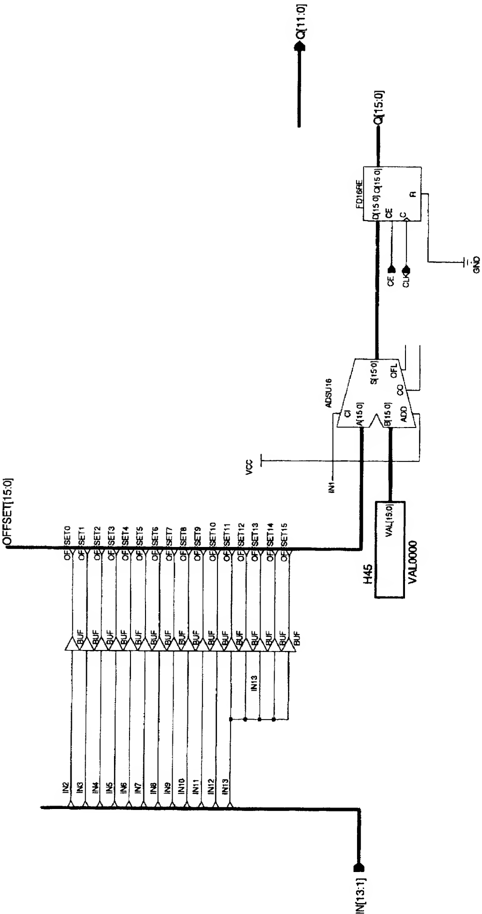

Q[12:0]
Q[15:0]
FD16RE
D[15:0]
Q[15:0]
CE
C
R
GND
CE
CLK
ADSU16
CI
A[15:0]
B[15:0]
ADD
S[15:0]
OFL
CO
VCC
IN2
OFFSET[15:0]
OFFSET0
OFFSET1
OFFSET2
OFFSET3
OFFSET4
OFFSET5
OFFSET6
OFFSET7
OFFSET8
OFFSET9
OFFSET10
OFFSET11
OFFSET12
OFFSET13
OFFSET14
OFFSET15
BUF
H45
VAL[15:0]
VAL0000
IN3
IN4
IN5
IN6
IN7
IN8
IN9
IN10
IN11
IN12
IN13
IN14
IN15
IN[15:2]

FIG. 35k



Q[11:0]
Q[15:0]
FD16RE
D[15:0]
Q[15:0]
CE
C
R
CE
CLK
GND
ADSU16
CI
A[15:0]
B[15:0]
ADD
S[15:0]
CO
OFL
VCC
IN1
OFFSET[15:0]
VAL[15:0]
H45
VAL0000
BUF
IN13
IN[13:1]

FIG. 35I

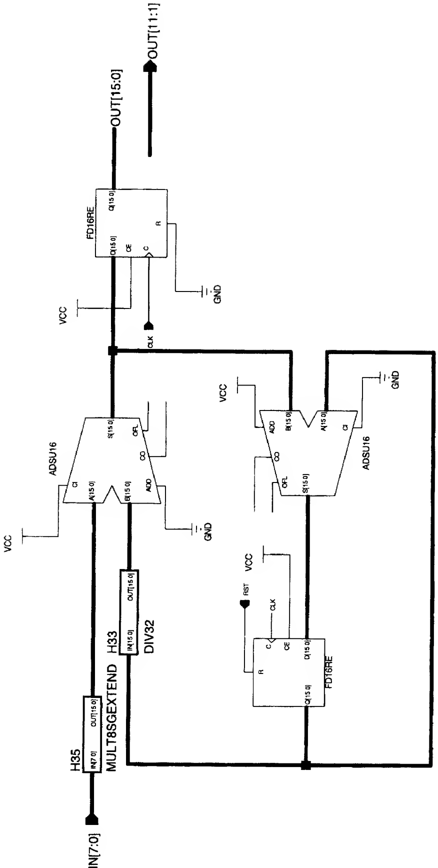

OUT[15:0]
OUT[11:1]
FD16RE
VCC
GND
CLK
ADSU16
ADSU16
VCC
GND
VCC
GND
H33
DIV32
H35
MULT8SGEXTEND
IN[7:0]
RST
CLK
VCC
FD16RE

FIG. 35m

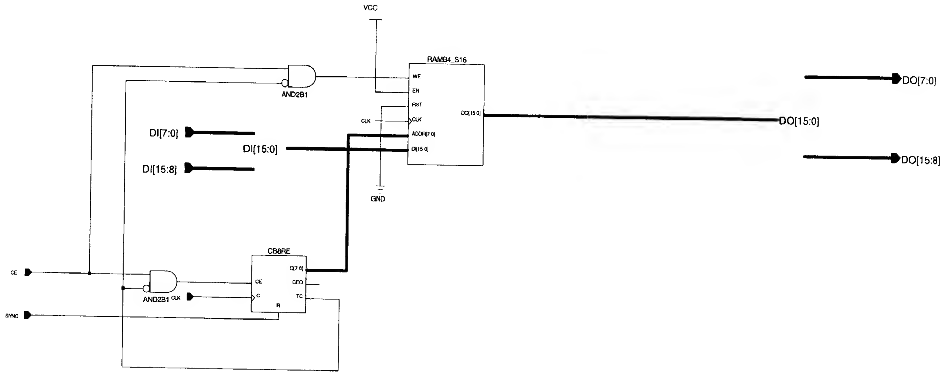

VCC
RAMB4_S16
WE
EN
RST
CLK
ADDR[7:0]
DI[15:0]
DO[15:0]
GND
AND2B1
DI[7:0]
DI[15:0]
DI[15:8]
DO[7:0]
DO[15:0]
DO[15:8]
CB8RE
Q[7:0]
CEO
TC
CE
C
R
AND2B1
CLK
CE
SYNC

FIG. 35n

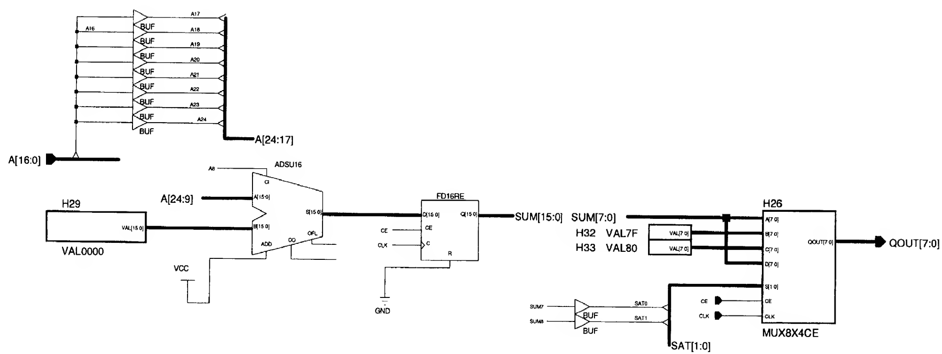

A16
A17
A18
A19
A20
A21
A22
A23
A24
BUF
A[24:17]
A[16:0]
A8
ADSU16
CI
A[15:0]
B[15:0]
S[15:0]
OFL
CO
ADD
H29
VAL[15:0]
VAL0000
A[24:9]
VCC
FD16RE
D[15:0]
Q[15:0]
CE
C
R
CLK
GND
SUM[15:0]
SUM[7:0]
H32 VAL7F
H33 VAL80
VAL[7:0]
H26
A[7:0]
B[7:0]
C[7:0]
D[7:0]
S[1:0]
QOUT[7:0]
MUX8X4CE
SUM7
SUM8
SAT0
SAT1
SAT[1:0]
QOUT[7:0]

FIG. 35o

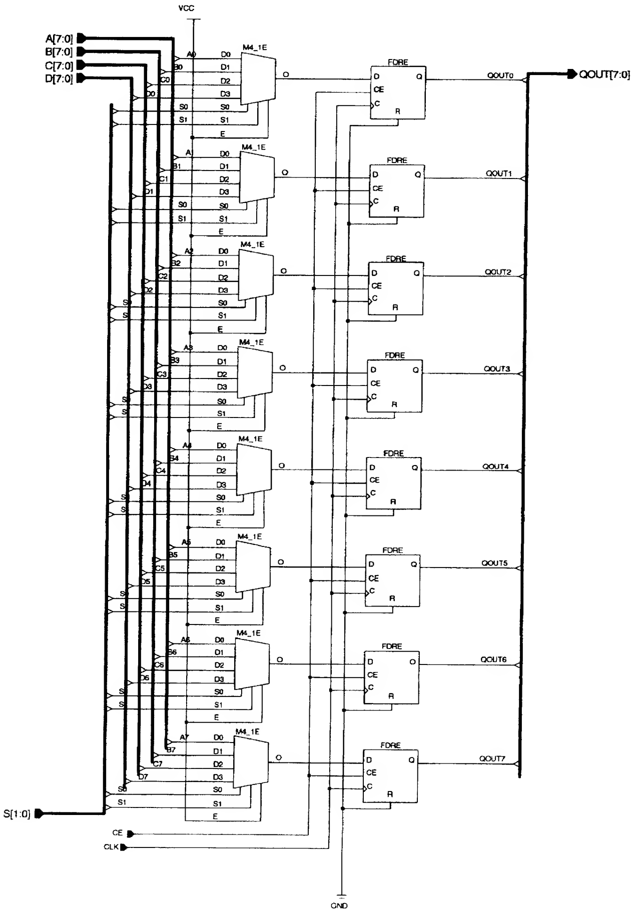
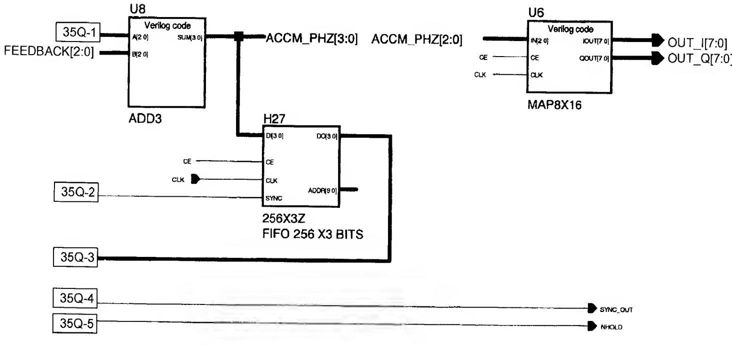

VCC
A[7:0]
B[7:0]
C[7:0]
D[7:0]
M4_1E
D0
D1
D2
D3
S0
S1
E
O
FDRE
D
CE
C
R
Q
QOUT0
QOUT1
QOUT2
QOUT3
QOUT4
QOUT5
QOUT6
QOUT7
QOUT[7:0]
S[1:0]
CE
CLK
GND

FIG. 35p



U8
Verilog code
A[2 0]
B[2 0]
SUM[3 0]
ADD3
35Q-1
FEEDBACK[2:0]
ACCM_PHZ[3:0]
ACCM_PHZ[2:0]
H27
DI[3 0]
DO[3 0]
CE
CLK
SYNC
ADDR[9 0]
256X3Z
FIFO 256 X3 BITS
35Q-2
35Q-3
35Q-4
35Q-5
U6
Verilog code
IN[2 0]
CE
CLK
IOUT[7 0]
QOUT[7 0]
MAP8X16
OUT_I[7:0]
OUT_Q[7:0]
SYNC_OUT
NHOLD

FIG. 35q-2

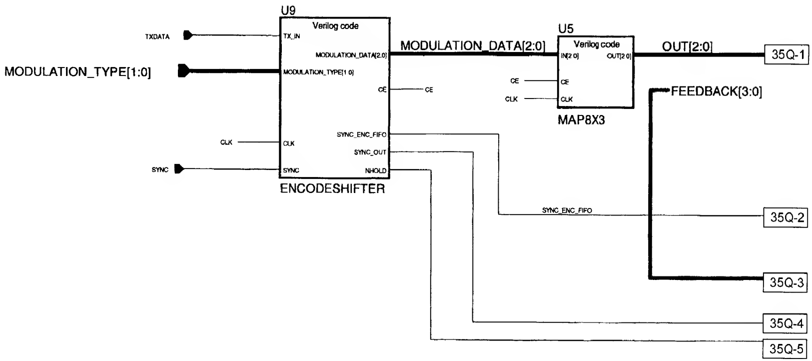

U9
Verilog code
TXDATA
TX_IN
MODULATION_TYPE[1:0]
MODULATION_TYPE[1 0]
MODULATION_DATA[2.0]
CE
CE
CLK
CLK
SYNC
SYNC
SYNC_ENC_FIFO
SYNC_OUT
NHOLD
ENCODESHIFTER
MODULATION_DATA[2:0]
U5
Verilog code
IN[2 0]
OUT[2 0]
CE
CE
CLK
CLK
MAP8X3
OUT[2:0]
FEEDBACK[3:0]
SYNC_ENC_FIFO
35Q-1
35Q-2
35Q-3
35Q-4
35Q-5

FIG. 35q-1

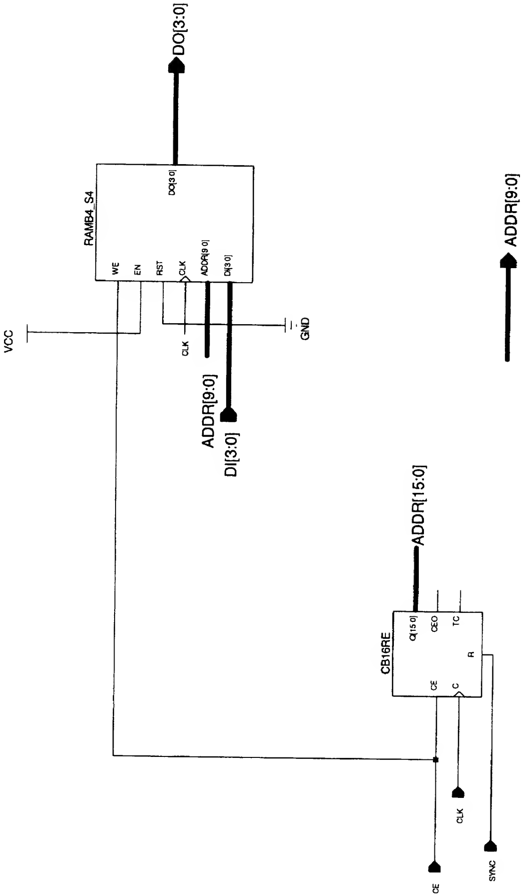

DO[3:0]
RAMB4_S4
DO[3:0]
WE
EN
RST
CLK
ADDR[9:0]
DI[3:0]
VCC
CLK
GND
ADDR[9:0]
DI[3:0]
ADDR[9:0]
ADDR[15:0]
CB16RE
Q[15:0]
CEO
TC
CE
C
R
CE
CLK
SYNC

FIG. 35r

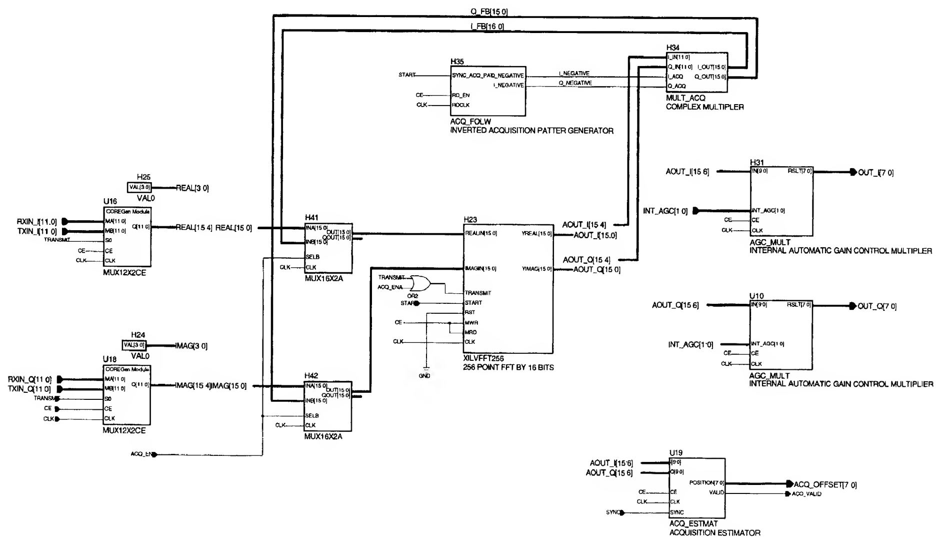

Q_FB[15 0]
I_FB[16 0]
H35
START
SYNC_ACQ_PAID_NEGATIVE
I_NEGATIVE
RD_EN
RDCLK
ACQ_FOLW
INVERTED ACQUISITION PATTER GENERATOR
H34
I_IN[11 0]
Q_IN[11 0]
I_ACQ
Q_ACQ
I_OUT[15 0]
Q_OUT[15 0]
Q_NEGATIVE
MULT_ACQ
COMPLEX MULTIPLER
H31
AOUT_I[15 6]
IN[9 0]
RSLT[7 0]
OUT_I[7 0]
INT_AGC[1 0]
AGC_MULT
INTERNAL AUTOMATIC GAIN CONTROL MULTIPLIER
U10
AOUT_Q[15 6]
OUT_Q[7 0]
AGC_MULT
INTERNAL AUTOMATIC GAIN CONTROL MULTIPLIER
H25
VAL[3 0]
REAL[3 0]
VAL0
U16
COREGen Module
RXIN_I[11.0]
TXIN_I[11 0]
TRANSMIT
MA[11 0]
MB[11 0]
Q[11 0]
S0
CE
CLK
REAL[15 4] REAL[15 0]
MUX12X2CE
H41
INA[15 0]
INB[15 0]
SELB
OUT[15 0]
QOUT[15 0]
MUX16X2A
H23
REALIN[15 0]
YREAL[15 0]
IMAGIN[15 0]
YIMAG[15 0]
AOUT_I[15 4]
AOUT_I[15.0]
AOUT_Q[15 4]
AOUT_Q[15 0]
TRANSMIT
ACQ_ENA
OR2
START
RST
MWR
MRD
GND
XILVFFT256
256 POINT FFT BY 16 BITS
H24
IMAG[3 0]
U18
RXIN_Q[11 0]
TXIN_Q[11 0]
IMAG[15 4]IMAG[15 0]
MUX12X2CE
H42
MUX16X2A
ACQ_EN
U19
AOUT_I[15 6]
AOUT_Q[15 6]
I[9 0]
Q[9 0]
POSITION[7 0]
VALID
SYNC
ACQ_OFFSET[7 0]
ACQ_VALID
ACQ_ESTMAT
ACQUISITION ESTIMATOR

FIG. 35s

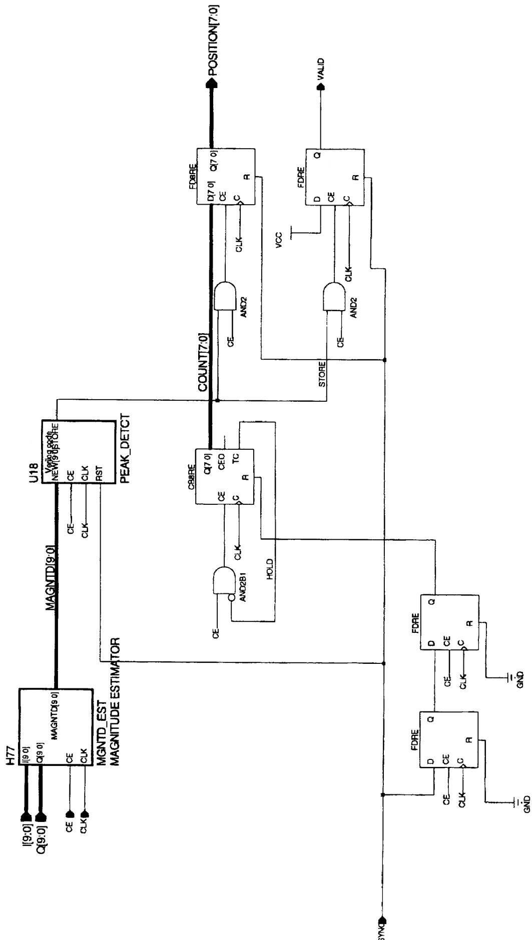

POSITION[7:0]
VALID
FD8RE
D[7:0]
Q[7:0]
CE
C
R
FDRE
D
Q
VCC
CLK
AND2
CE
COUNT[7:0]
STORE
U18
Verilog code
NEW[9:0]
STORE
RST
PEAK_DETCT
CB8RE
CEO
TC
AND2B1
HOLD
MAGNTD[9:0]
H77
I[9:0]
Q[9:0]
MGNTD_EST
MAGNITUDE ESTIMATOR
GND
SYNC

FIG. 35t

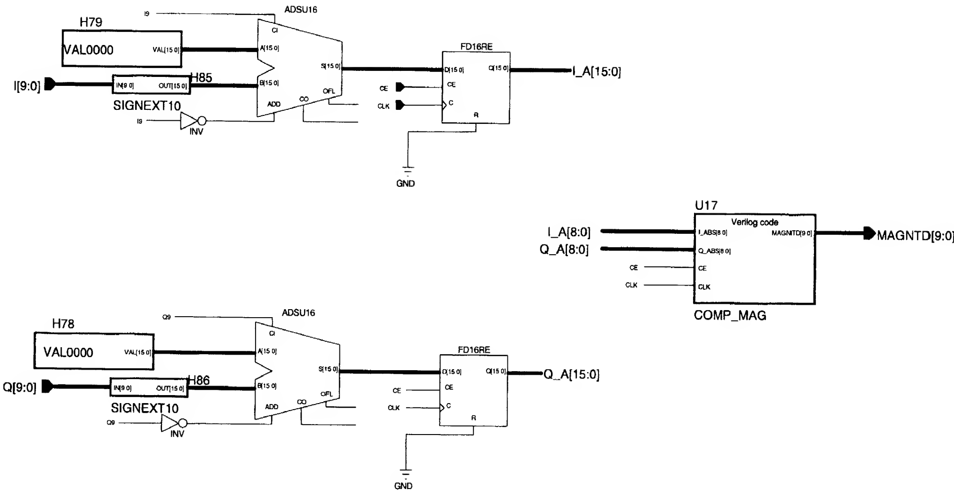

H79
VAL0000
VAL[15 0]
I9
ADSU16
CI
A[15 0]
B[15 0]
ADD
CO
OFL
S[15 0]
I[9:0]
IN[9 0]
OUT[15 0]
H85
SIGNEXT10
I9
INV
FD16RE
D[15 0]
Q[15 0]
CE
C
R
CE
CLK
GND
I_A[15:0]
U17
Verilog code
I_ABS[8 0]
Q_ABS[8 0]
CE
CLK
MAGNTD[9 0]
COMP_MAG
I_A[8:0]
Q_A[8:0]
CE
CLK
MAGNTD[9:0]
H78
VAL0000
VAL[15 0]
Q9
ADSU16
CI
A[15 0]
B[15 0]
ADD
CO
OFL
S[15 0]
Q[9:0]
IN[9 0]
OUT[15 0]
H86
SIGNEXT10
Q9
INV
FD16RE
D[15 0]
Q[15 0]
CE
C
R
CE
CLK
GND
Q_A[15:0]

FIG. 35u

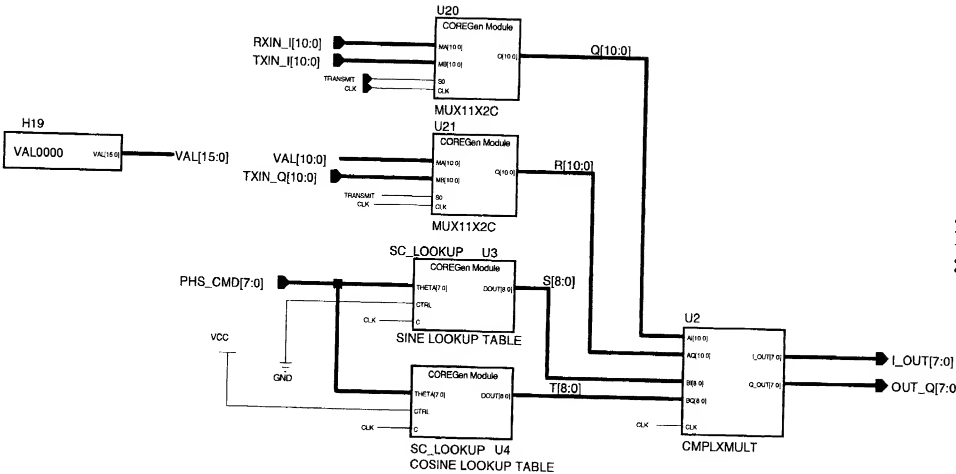

U20
COREGen Module
RXIN_I[10:0]
TXIN_I[10:0]
TRANSMIT
CLK
Q[10:0]
MUX11X2C
H19
VAL0000
VAL[15:0]
U21
COREGen Module
VAL[10:0]
TXIN_Q[10:0]
TRANSMIT
CLK
R[10:0]
MUX11X2C
SC_LOOKUP U3
COREGen Module
PHS_CMD[7:0]
S[8:0]
CLK
SINE LOOKUP TABLE
VCC
GND
COREGen Module
T[8:0]
CLK
SC_LOOKUP U4
COSINE LOOKUP TABLE
U2
I_OUT[7:0]
OUT_Q[7:0]
CLK
CMPLXMULT

FIG. 35V

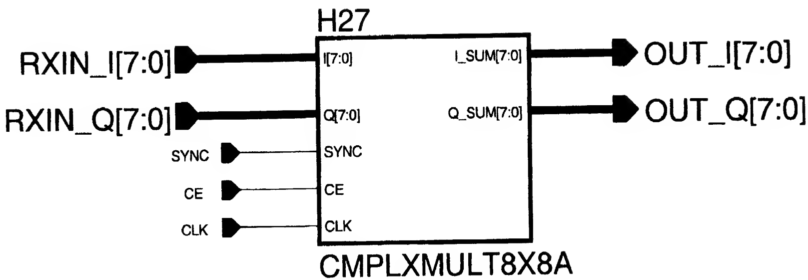

H27
RXIN_I[7:0]
RXIN_Q[7:0]
SYNC
CE
CLK
I[7:0]
Q[7:0]
SYNC
CE
CLK
I_SUM[7:0]
Q_SUM[7:0]
OUT_I[7:0]
OUT_Q[7:0]
CMPLXMULT8X8A

FIG. 35w

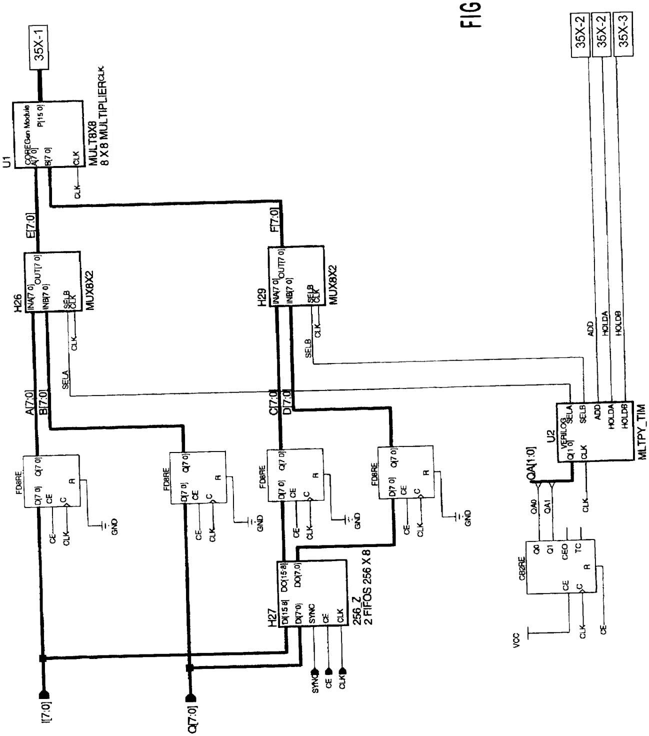

35X-1
U1
COREGen Module
A[7:0]
B[7:0]
P[15:0]
CLK
MULT8X8
8 X 8 MULTIPLIER
E[7:0]
F[7:0]
H26
H29
INA[7:0]
INB[7:0]
OUT[7:0]
SELB
SELA
MUX8X2
A[7:0]
B[7:0]
C[7:0]
D[7:0]
FD8RE
D[7:0]
Q[7:0]
CE
C
R
GND
H27
DI[15:8]
DI[7:0]
DO[15:8]
DO[7:0]
SYNC
256_Z
2 FIFOS 256 X 8
I[7:0]
Q[7:0]
U2
VERILOG
Q[1:0]
ADD
HOLDA
HOLDB
MLTPY_TIM
QA[1:0]
QA0
QA1
CB2RE
Q0
Q1
CEO
TC
VCC
35X-2
35X-2
35X-3

FIG. 35x-1

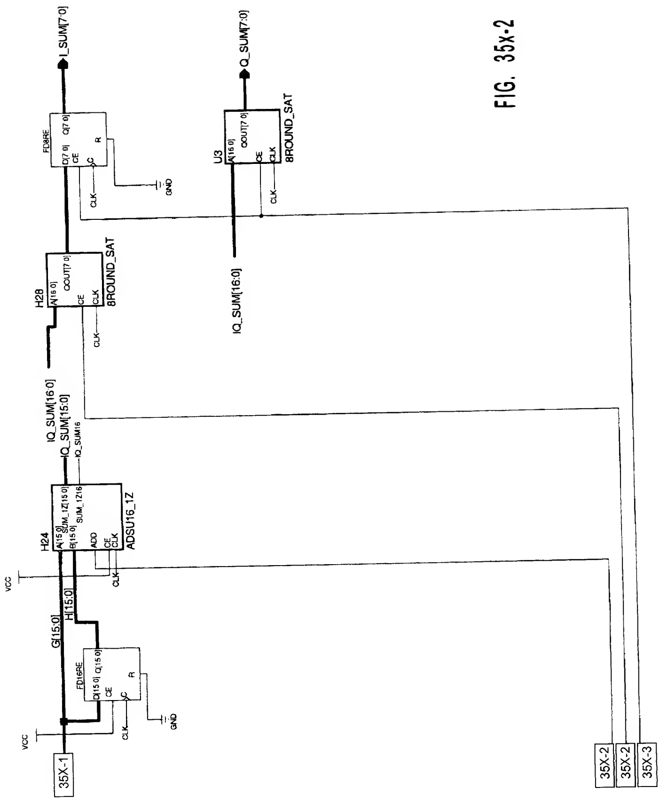

35X-1
VCC
CLK
GND
FD16RE
D[15:0]
Q[15:0]
CE
C
R
G[15:0]
H[15:0]
H24
A[15:0]
B[15:0]
SUM_1Z[15:0]
SUM_1Z16
ADD
CE
CLK
ADSU16_1Z
IQ_SUM[16:0]
IQ_SUM[15:0]
IQ_SUM16
H28
A[16:0]
OOUT[7:0]
8ROUND_SAT
FD8RE
D[7:0]
Q[7:0]
I_SUM[7:0]
IQ_SUM[16:0]
U3
Q_SUM[7:0]
35X-2
35X-2
35X-3

FIG. 35x-2

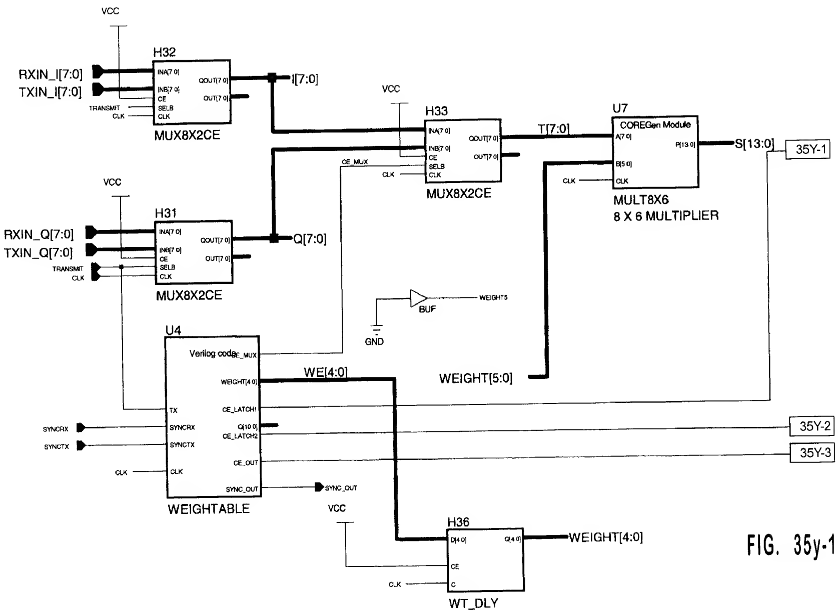

VCC
H32
RXIN_I[7:0]
TXIN_I[7:0]
INA[7 0]
INB[7 0]
CE
SELB
CLK
QOUT[7 0]
OUT[7 0]
TRANSMIT
CLK
MUX8X2CE
I[7:0]
VCC
H33
INA[7 0]
INB[7 0]
CE
SELB
CLK
QOUT[7 0]
OUT[7 0]
CE_MUX
CLK
MUX8X2CE
T[7:0]
U7
COREGen Module
A[7 0]
B[5 0]
CLK
P[13 0]
S[13:0]
35Y-1
CLK
MULT8X6
8 X 6 MULTIPLIER
VCC
H31
RXIN_Q[7:0]
TXIN_Q[7:0]
INA[7 0]
INB[7 0]
CE
SELB
CLK
QOUT[7 0]
OUT[7 0]
TRANSMIT
CLK
MUX8X2CE
Q[7:0]
BUF
WEIGHT5
GND
U4
Verilog code
CE_MUX
WEIGHT[4 0]
TX
CE_LATCH1
SYNCRX
Q[10 0]
CE_LATCH2
SYNCTX
CE_OUT
CLK
SYNC_OUT
SYNCRX
SYNCTX
CLK
WEIGHTABLE
WE[4:0]
WEIGHT[5:0]
35Y-2
35Y-3
SYNC_OUT
VCC
H36
D[4 0]
Q[4 0]
CE
C
CLK
WT_DLY
WEIGHT[4:0]

FIG. 35y-1

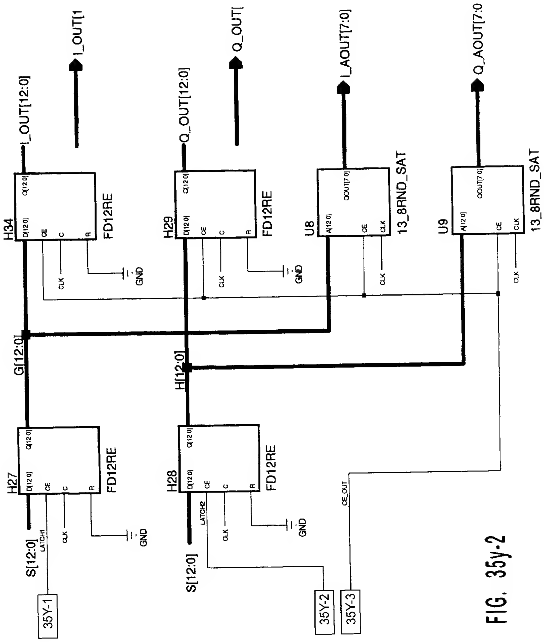

I_OUT[12:0]
I_OUT[1
Q_OUT[12:0]
Q_OUT[
I_AOUT[7:0]
Q_AOUT[7:0
H34
FD12RE
H29
FD12RE
U8
13_8RND_SAT
U9
13_8RND_SAT
G[12:0]
H[12:0]
H27
FD12RE
H28
FD12RE
S[12:0]
LATCH1
S[12:0]
LATCH2
CE_OUT
CLK
GND
35Y-1
35Y-2
35Y-3

FIG. 35y-2

PHASE ACCUMULATOR

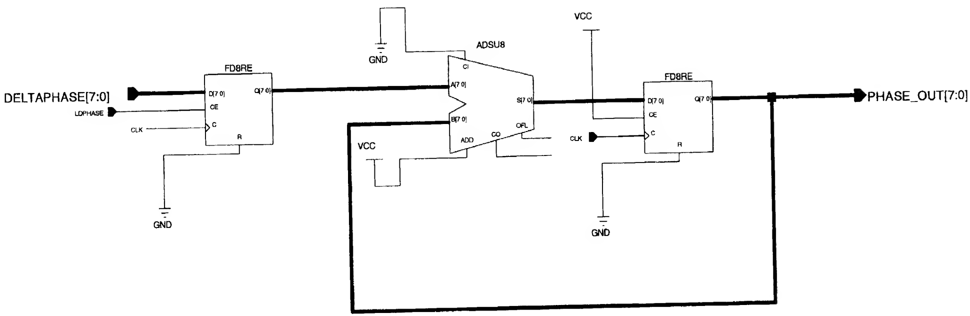


FIG. 35z

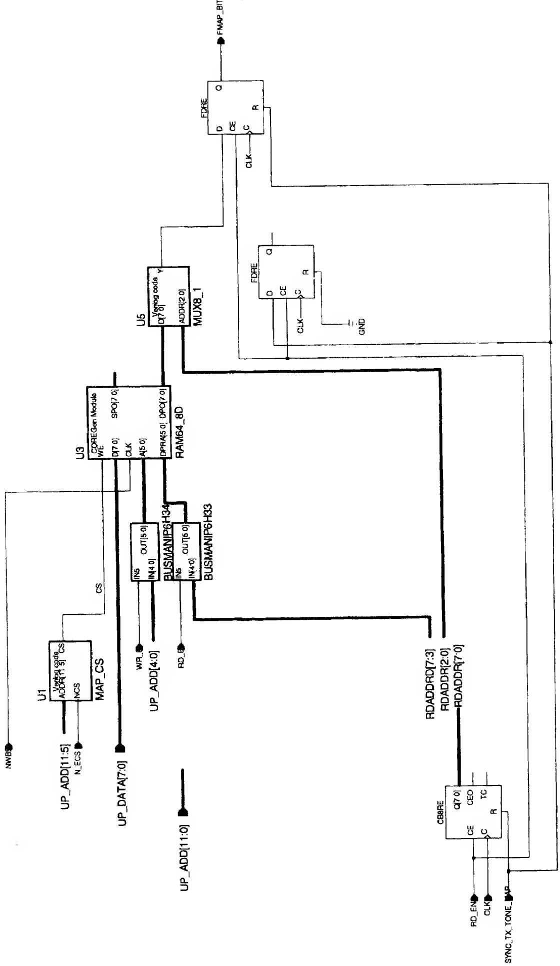

FMAP_BIT
FDRE
D
CE
C
Q
R
CLK
FDRE
D
CE
C
Q
R
CLK
GND
U5
Verilog code
D[7:0]
ADDR[2:0]
Y
MUX8_1
U3
COREGen Module
WE
D[7:0]
CLK
A[5:0]
DPRA[5:0]
SPO[7:0]
DPO[7:0]
RAM64_8D
IN5
IN[4:0]
OUT[5:0]
BUSMANIP6H34
IN5
IN[4:0]
OUT[5:0]
BUSMANIP6H33
CS
U1
Verilog code
ADDR[11:5]
NCS
CS
MAP_CS
NWBE
UP_ADD[11:5]
N_ECS
UP_DATA[7:0]
WR_E
UP_ADD[4:0]
RD_E
UP_ADD[11:0]
RDADDRD[7:3]
RDADDR[2:0]
RDADDR[7:0]
CB8RE
CE
C
Q[7:0]
CEO
TC
R
RD_EN
CLK
SYNC_TX_TONE_MAP

FIG. 35aa

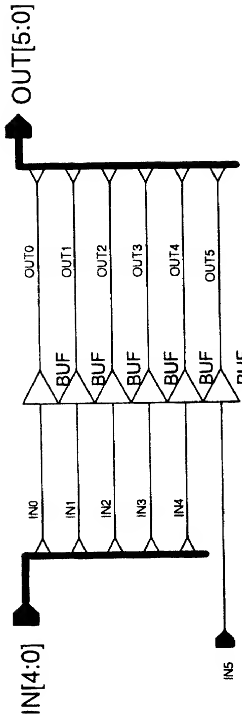

OUT[5:0]
OUT0
OUT1
OUT2
OUT3
OUT4
OUT5
BUF
BUF
BUF
BUF
BUF
BUF
IN0
IN1
IN2
IN3
IN4
IN[4:0]
IN5

FIG. 35ab

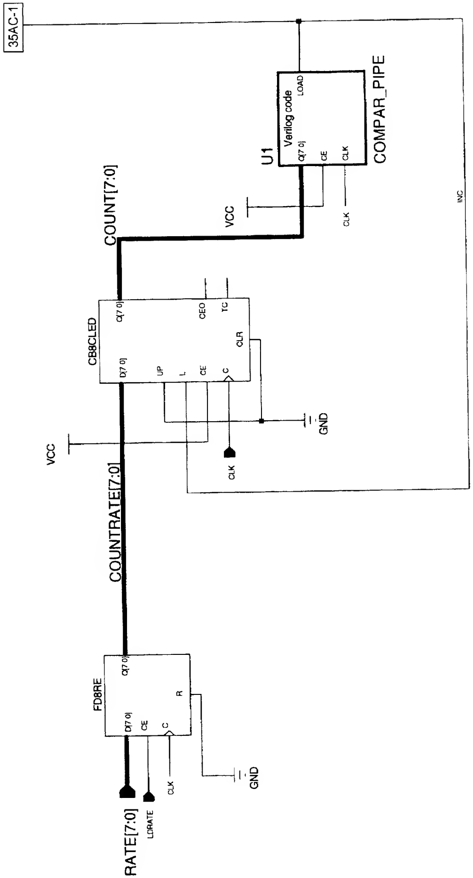

35AC-1
U1
Verilog code
Q[7:0]
LOAD
CE
CLK
COMPAR_PIPE
COUNT[7:0]
VCC
CLK
INC
CB8CLED
D[7:0]
Q[7:0]
UP
L
CE
C
CEO
TC
CLR
GND
VCC
CLK
COUNTRATE[7:0]
FD8RE
D[7:0]
Q[7:0]
CE
C
R
GND
CLK
RATE[7:0]
LDRATE

FIG. 35ac-1

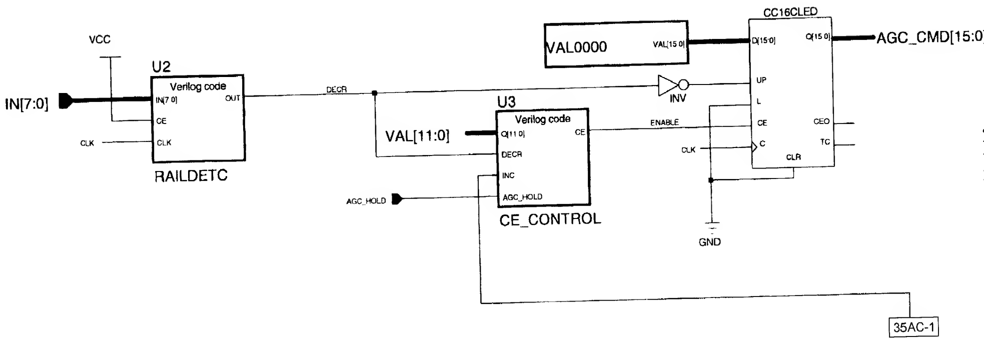

VCC
U2
Verilog code
IN[7:0]
IN[7 0]
OUT
CE
CLK
CLK
RAILDETC
DECR
VAL[11:0]
U3
Verilog code
Q[11 0]
CE
DECR
INC
AGC_HOLD
AGC_HOLD
CE_CONTROL
VAL0000
VAL[15 0]
INV
ENABLE
CLK
GND
CC16CLED
D[15:0]
Q[15 0]
UP
L
CE
C
CLR
CEO
TC
AGC_CMD[15:0]
35AC-1

FIG. 35ac-2

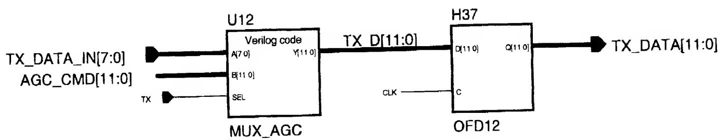

U12
Verilog code
A[7 0]
B[11 0]
SEL
Y[11 0]
MUX_AGC
TX_DATA_IN[7:0]
AGC_CMD[11:0]
TX
TX_D[11:0]
H37
D[11 0]
Q[11 0]
C
CLK
OFD12
TX_DATA[11:0]

FIG. 35ac-3

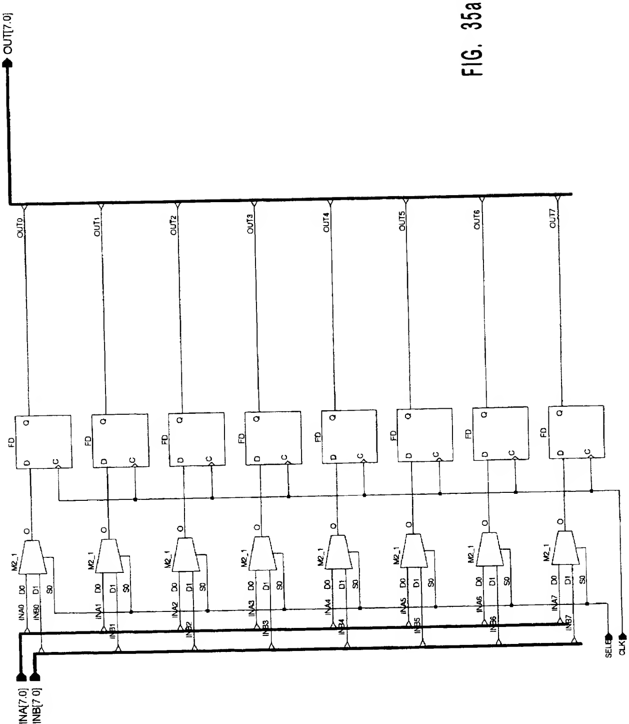

OUT[7:0]
OUT0
OUT1
OUT2
OUT3
OUT4
OUT5
OUT6
OUT7
FD
D
C
Q
M2_1
D0
D1
S0
O
INA0
INB0
INA1
INB1
INA2
INB2
INA3
INB3
INA4
INB4
INA5
INB5
INA6
INB6
INA7
INB7
INA[7:0]
INB[7:0]
SELB
CLK

FIG. 35ad

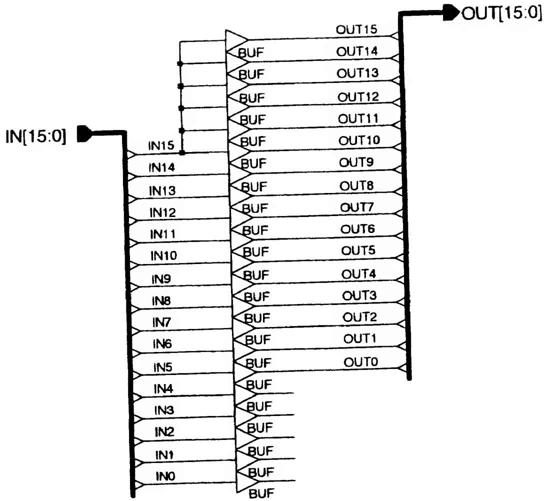

IN[15:0]
OUT[15:0]
IN15
IN14
IN13
IN12
IN11
IN10
IN9
IN8
IN7
IN6
IN5
IN4
IN3
IN2
IN1
IN0
OUT15
OUT14
OUT13
OUT12
OUT11
OUT10
OUT9
OUT8
OUT7
OUT6
OUT5
OUT4
OUT3
OUT2
OUT1
OUT0
BUF

FIG. 35ae

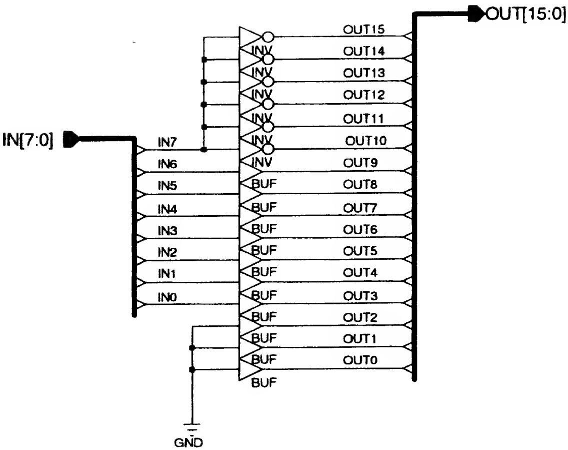

IN[7:0]
OUT[15:0]
IN7
IN6
IN5
IN4
IN3
IN2
IN1
IN0
INV
BUF
OUT15
OUT14
OUT13
OUT12
OUT11
OUT10
OUT9
OUT8
OUT7
OUT6
OUT5
OUT4
OUT3
OUT2
OUT1
OUT0
GND

FIG. 35af